REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
31-03-2017	Final	24-09-2012 – 23-09-2016

4. TITLE AND SUBTITLE

Large Scale Data Analysis and Knowledge Extraction in Communication Data

5a. CONTRACT NUMBER
W911NF-12-2-0067

5b. GRANT NUMBER
W911NF-12-2-0067

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Dr. Roy George

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Clark Atlanta University
223 James P. Brawley Dr., SW
Atlanta, Georgia 30314

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory

10. SPONSOR/MONITOR'S ACRONYM(S)

ARL

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Proper evaluation of technology in operational settings is a vital to an optimal selection of equipment and technology. For the US Army the selection of communication equipment is vital in the battlefield and an in depth understanding of performance characteristics is essential. Operational data analysis of data is an approach which can reveal performance characteristics in the real world. However, a moderate number of equipment can produce a large volume of data gathered, with numerous variables, and complex interdependencies making it impossible for conventional data analysis. These considerations also require massive scale up needed to tackle the "big" data component adding to several magnitudes of difficulty to the problem. The objective of this research and development effort is to develop techniques for (i) efficient information acquisition and movement through the parallel environment, (ii) generate metrics from that data that promotes understanding of the basic features of the data, and (iii) development of frequent pattern mining algorithms to understand the interdependencies of the data parameters. This would provide technology analysts with quick answers to questions about system performance in real world scenarios. A second outcome of this research work would be the methodical development of a software infrastructure that will permit analysis of the data, exploiting massively parallelized versions of advanced data mining algorithms, needed to understand the complex relationships and dependencies in a blended selection of real and synthetic "big" data.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Dr. Roy George
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	**UNCLASSIFIED, UNLIMITED (U2) PUBLIC RELEASE**	43	**19b. TELEPHONE NUMBER** *(include area code)*
UU	UU	UU			404/880-6945

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

CLARK ATLANTA UNIVERSITY

OFFICE OF THE VICE PRESIDENT FOR
RESEARCH AND SPONSORED PROGRAMS

April 27, 2016

Mr. Robert Sheroke
U.S. Army Research Laboratory
RDRL-CIH (Mr. Robert Sheroke)
Building 120, Room O318
Aberdeen Proving Ground, MD 21005

Subject: Final Report: Contract/Grant Number W911NF-12-2-0067
Large Scale Data Analysis and Knowledge Extraction in Communication Data; Dr. Roy George, Principal Investigator

Dear Mr. Sheroke:

Enclosed are the original and one copy of the final report for contract/grant number W911NF-12-2-0067 issued to Clark Atlanta University.

Additionally, enclosed is DD Form 882.

Thank you for the opportunity to work with the U.S. Army Research Laboratory.

Sincerely,

Carol E Johnson

Carol E. Johnson (Ms.)
Assistant Vice President
Research, Sponsored Programs and
Dual Degree Engineering

Enclosures

223 JAMES P. BRAWLEY DRIVE S.W. • ATLANTA, GEORGIA 30314-4391 • TEL: (404) 880-6990 • FAX: (404) 880-6983
RESEARCH@CAU.EDU • WWW.CAU.EDU/RESEARCH

Contract No: W911NF-12-2-0067

Title: Large Scale Data Analysis and Knowledge Extraction in Communication Data

Final Report

Background:

The analysis of communication equipment characteristics in operation settings is a vital component of optimal equipment and technology selection. For the US Army in depth analysis of the equipment performance and efficiencies are vital to ensuring that these technologies work as expected in the battlefield. In depth understanding of performance characteristics in a real world situation has to be understood in a context where actual data is blended with simulation data. We propose that this problem be tackled in three phases. Initially this involves acquisition of the data, cleaning and normalization. It will also incorporate familiarization with the parallel environment on the test ARL hardware platform. The second phase of this effort involves extraction of key data metrics that will help in understanding of the data, the relationships between data parameters and resolving the structural and semantic conflicts within the data. The third phase of this project will build on the first two phases and develop techniques that will be used to analyze and discover latent associations within the data.

Work Accomplished:

During this reporting period, we have expanded on the approach taken in Year 1 on network characterization. Using the network characterization, we have developed methods to determine network performance and rank the network, directly meeting the objectives of this grant. In related work, we have also developed measures to find outliers in networks using the notions of betweeness. The details of the work performed are summarized below:

- **Net Performance Rank: A Comparison Measure to Determine Network Performance Ranking**

  Here we consider the basic approach to the problem of determining the quality of the equipment used in communication. The network tested can be of any topology and the while the formulations do not consider physical factors; these may be introduced through appropriate weightage of parameters. This problem is a ranking problem in Multi-Criteria Decision Making (MCDM) and the approach is based on previous work performed under this contract . A MCDM problem can be expressed in matrix format as

	C_1	C_2	...	C_n
A_1	x_{11}	x_{12}	...	x_{1n}
A_2	x_{21}	x_{22}	...	x_{2n}
A_m	x_{m1}	x_{m2}	...	x_{mn}

$$W = [w_1, w_2, \ldots, w_n]$$

Where $A_1, A_2, \ldots, A_m$ are possible alternative networks among which have to rank, $c_1, c_2, \ldots, c_n$ are criteria with which alternative performance are measured, x_{ij} is the score of alternative A_i with respect to criterion c_j, w_j is the weight of criterion c_j.

While various attributes may be selected as evaluating criteria we focus on positive/negative frequent events which may occur between nodes during a given period of time, and effect on network's performance. For instance, the retransmissions or other failures are negative events that decrease network's efficiency. To establish the decision matrix, the follow steps needed: 1.Select the collection of criteria, and 2. Scoring networks on criteria.

With the purpose of scoring networks on criteria, we proposed a new density based approach which compute global probability density of the given positive/negative frequent event (criteria) for each networks. For this purpose, we developed a novel method the "*Correlation Density Rank*" which finds probability density distribution of related frequent event on all nodes, and then we aggregate these densities on whole network using the Renyi entropy as the score of network performance on related criterion. The Topsis method is then used to calculate the performance rank of the network.

- **Discovering Community Structure in Dynamic Networks**

Recent studies have supplied favorable results regarding to exploring communities within a dynamic networks, a major problem in the data mining area. A correct community is usually defined as a subgraph with a higher internal density and a lower crossing density with others subgraphs. Various density-based techniques have been devoted to uncovering community structures in social networks. In this research effort, a novel distance based ranking algorithm, which is called "Correlation Density Rank", is developed to derive the community tree from the network. As in the real world, where a network is constantly evolving, we demonstrate a tree learning algorithm, which employs edit distance as the scoring function, to derive an evolving community tree that allows a smooth alteration between two community trees. We also string communities to obtain an evolution graph of the organizational structure, by which we can achieve new perceptions from the

dynamic network. The experiments, conducted on a synthetic graph and the real-world network dataset provided by ARL demonstrate the feasibility and applicability of the framework.

- **Outlier Detection in Network Data using the Betweenness Centrality**

  Outlier detection has been used to detect and, where appropriate, remove anomalous observations from data. It has important applications in the field of fraud detection, network robustness analysis, and intrusion detection. In this approach, we propose Betweenness Centrality as a technique to determine the outlier in network analyses. The Betweenness Centrality of a vertex in a graph is a measure for the participation of the vertex in the shortest paths in the graph. This measure is widely used in network analyses where the recursive computation of the betweenness centralities of vertices is used to for community detection. We show the effectiveness of using this method to detect outliers in network data.

- **A Smart Assignment Technique with Consideration of Multicriteria Reciprocal Judgments**

  To date the assignment problems are important tasks in recommender systems and one-to-one matching issues through social environments. The various approaches have been proposed to reach these purposes that are normally limited to the considerations of cost or profit incurred by each possible assignment. However most of the time, each of the alternatives at both assignment sides have particular criteria for judging about the other side alternatives, whereby they can evaluate their sufficiency. In this paper, in order to obtain the optimality of both dimensions of assignment we try to consider the concept of efficiency rather than the cost or profit of each possible assignment. Therefore, the efficient assignment is the one that firstly, has the maximum optimality in terms of both dimensions of assignment, and secondly, takes into account the significance of judgment of each assignment from the viewpoint of decision maker. To do this, a compound index would be defined which includes the efficiency related to twodimensional optimized assignment for the purpose of measuring the performance of each possible assignment. Next, A mathematical programming model for the extended assignment problem is proposed, which is then expressed as a classical integer linear programming model to determine the assignments with the maximum efficiency. A numerical example is used to demonstrate the approach.

Technical Papers:

Using the funding from this grant, the CAU team has developed the following papers:

H. B. Mihiri Shashikala, Roy George, Hsin-Chu Chen, "Outlier Detection in Network Data using the Betweenness Centrality," *Proceedings of the IEEE Southeastcon,* Fort Lauderdale, April 2015

Z. Bahrami Bidoni, R.George and A. Makui, "A Smart Assignment with Consideration of Multicriteria Reciprocal Judgments", *2014 ASE BigData/ SocialInformatics/ PASSAT/ BioMedCom 2014 Conference,* Harvard University, December 14-16, 2014.

Z. Bahrami Bidoni, R.George, "Network Performance Rank: An Approach for Comparison of Complex Networks," *2014 ASE BigData/ SocialInformatics/ PASSAT/ BioMedCom 2014 Conference*, Harvard University, December 14-16, 2014.

Z. Bahrami Bidoni, R.George,"Network Service Quality Rank: A Network Selection Algorithm for Heterogeneous Wireless Networks," *Proceedings of the Tenth ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, pp.239- 240. ACM, 2014.

Z. Bahrami Bidoni and R.George, "Discovering Community Structure in Dynamic Social Networks using the Correlation Density Rank," *2014 ASE BigData/SocialCom/Cybersecurity Conference*, Stanford University, May 27-31, 2014.

Z. Bahrami Bidoni, R. George, K.A. Shujaee, "A Generalization of the PageRank Algorithm," *Proceedings of The Eighth International Conference on Digital Society (ICDS),* April 2014

Students Supported

Isaiah Grigsby, Undergraduate Student, Male (currently enrolled)

Tabiah Conrad, Undergraduate Student, Female (currently enrolled)

B. Williams, Graduate Student, Male (at Boeing Corp.)

M. Brooks, Undergraduate Student, Male (startup company, Purview L.L.C.)

S. Reed, Undergraduate Student, Female (Graduate School, Georgia Tech.)

K. Robinson, Undergraduate Student, Male (at Ernst & Young)

R. Silva, Graduate Student, Female (at Morehouse School of Medicine)

Z. Bidoni, Graduate Student, Female (Ph. D. Program, Georgia Tech.)

H. Shashikala, Graduate Student, Female (Ph. D. Program, Clemson University)

O. ElTabeby, Graduate Student, Male (Ph. D. Program, University of North Carolina)

Publications

Outlier Detection in Network Data using the Betweenness centrality

H. B. Mihiri Shashikala, Roy George, Khalil A. Shujaee
Department of Computer and Information Science
Clark Atlanta University
Atlanta, GA 30314
hewa.shashikala@students.cau.edu

Abstract— **Outlier detection has been used to detect and, where appropriate, remove anomalous observations from data. It has important applications in the field of fraud detection, network robustness analysis, and intrusion detection. In this paper, we propose a Betweenness Centrality (BEC) as novel to determine the outlier in network analyses. The Betweenness Centrality of a vertex in a graph is a measure for the participation of the vertex in the shortest paths in the graph. The Betweenness centrality is widely used in network analyses. Especially in a social network, the recursive computation of the betweenness centralities of vertices is performed for the community detection and finding the influential user in the network. In this paper, we propose that this method is efficient in finding outlier in social network analyses. Furthermore we show the effectiveness of the new methods using the experiments data.**

Keywords—outlier detection; network data; betweenness centrality, adjecncy matrix.

I. INTRODUCTION

Outlier detection is an important data mining task that is focused on the discovery of objects that are exceptional when compared with a set of observations that are considered typical. In many data analysis tasks, a large number of variables are being recorded or sampled. One of the first steps towards obtaining a coherent analysis is the detection of outlaying observations. Although outliers are often considered as an error or noise, they may carry important information. Detected outliers are candidates for aberrant data that may otherwise adversely lead to model misspecification, biased parameter estimation and incorrect results. These objects are important since they often lead to the discovery of exceptional events. Substantial research has been done in outlier detection and these are classified into different types with respect to the detection approach being used. Exemplar techniques include Classification based methods, Nearest Neighbor based methods, Cluster based methods and Statistical based methods [19]. In the Classification-based approach [31], [32] a model is created from a set of labeled data points and then a test point is classified into one of the classes using appropriate testing. Support Vector Machine (SVM) based methods [30], methods based on Neural Networks [33] and Bayesian Networks based methods [25],[28],[34] belong to Classification based technique. The testing phase of this method is considerably fast as each test data is compared against the pre-built model. The accuracy of classification based methods rely on the availability of accurate pre classified examples for different normal classes, which is rarely found. Nearest Neighbor based methods [27], [29], [35] involve distance or similarity measures which is defined between data points. In this paper, we discuss a new method to find out an outlier that is based on a graph. This method efficiently reduces the search space by finding a candidate set of vertices whose betweenness centralities can be computed using candidate vertices only.

The Betweenness Centrality (BEC) is a measure that computes the relative importance of a vertex in a graph, and it is widely used in network analyses such as social network analysis, biological graph analysis, and road network analysis [1]. In the social network analysis, a vertex with higher centrality can be viewed as a more important vertex than a vertex with lower centrality. The BEC of a vertex in a graph is a measure used for the participation of the vertex in the shortest paths in the graph. There are many previous works on the BEC problem. The concept of the BEC is proposed in [35], but the definition proposed in [40] is more widely used. Recently, many variants of the definition are proposed in [38], [37] improves the computation time of the BEC based on a modified breadth-first search algorithm and the dependency of a vertex, and it is the fastest known algorithm that computes the exact BEC of all the vertices in a graph. The computations of the shortest paths between all pairs of vertices are time consuming. Therefore, another definition of BEC is proposed [22]; this based on a random walk. In [42], each vertex has a probability of visiting its neighbor vertices. Also, [39], [36] and [41] propose approximation algorithms for computing the betweenness centrality. [43] and [44] adopt the betweenness centrality for detecting communities in a social network.

Although many methods currently exist on calculating the BEC and the BEC is one of the major methods used in analyzing social network graphs, none of the existing methods address the problem of updating BEC. In this paper we propose the betweenness centrality to find out outliers for network type data.

The next section of this paper describes related terms and definitions which are used throughout the paper. Furthermore, it outlines the approach that explains the algorithm behind the BEC approach. To get a better understanding and to demonstrate the accuracy of BEC, several

experiments were conducted with different kinds of synthetic data sets which are described in detail in the experimental results section. We apply BEC technique to find outliers in synthetic data sets and compare it with another an alternate technique the modified-Shared Nearest Neighbor[3]. Finally we conclude the paper with a discussion of the performance, accuracy and the importance of the proposed technique. From the results of experiments, it is clear that this technique gives better results in comparison to the modified-Shared Nearest Neighbor by giving higher true positive and true negative values and very low false positive and false negative values for network type data.

The m-SNN (modified-Shared Nearest Neighbor) method [3] is based on the non-parametric clustering algorithm, the Shared Nearest Neighbor (SNN) Approach developed by Ertöz et al. [9]. This method, we consider the ratio between the summation of Euclidean distances to shared nearest neighbors and their total number of shared neighbors. To differentiate between outliers and normal nodes, hypothesis testing is used, Babara et al [18] and Rogers [4].

II. Terms and Definitions

Betweenness Centrality

A measure that computes the relative importance of a vertex in a graph. The formal definition is presented below.

A graph is represented by $G=(V,E)$, where V is the set of vertices, and $E \subseteq V \times V$ is the set of edges. A path in a graph is represented by a sequence of vertices, $(v_1,...,v_n)$ where $v_i, v_j \in V$ for $1 \le i, j \le n$, $i \ne j$, except possible $1=n$.

Definition 1 (Betweenness Centrality). The betweenness centrality of a vertex $v_j \in G$ is:

$$c(v_j)=\sum_{i,k}\frac{\sigma_{v_i,v_k}(v_j)}{\sigma_{v_i,v_k}(1)}$$

Where, $v_i, v_j, v_k \in V$, $i \ne j \ne k$, $\sigma_{v_i,v_k}(v_j)$ is the number of shortest paths between v_i and v_k that include v_j, and σ_{v_i,v_k} is the number of shortest paths between v_i and v_k. The betweenness centrality can be computed as follows:

1 For each pair of vertices (v_s and v_t), compute the shortest paths between the two vertices.

2. For each pair of vertices, compute the ratio of each vertex participating in the shortest path(s). The ratio is the number of shortest paths between v_s and v_t that go through v_j divided by the number of shortest paths between v_s and v_t.

3. Accumulate the ratio for all pairs of vertices.

Definition 2 (Adjacency Matrices).

The adjacency matrix of a finite graph G on n vertices is the $n \times n$ matrix where the non-diagonal entry a_{ij} is the number of edges from vertex i to vertex j, and the diagonal entry a_{ii}, depending on the convention, is either once or twice the number of edges (loops) from vertex i to itself. Undirected graphs often use the latter convention of counting loops twice, whereas directed graphs typically use the former convention.

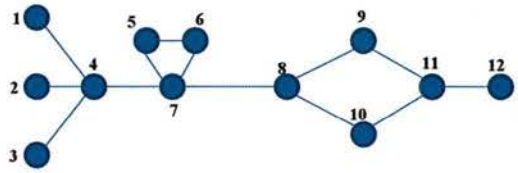


Figure 1: Shortest paths through nodes in destination IP addresses.

	A	B	C	D	E	F
A	0	1	1	1	1	-
B	1	0	-	1	1	-
C	1	-	0	-	-	1
D	1	1	-	0	-	1
E	1	1	-	-	0	1
F	-	-	1	1	1	0

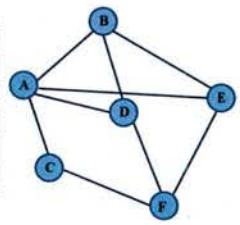


Figure 2:Undirected graph with adjacency matrix.

Figure 2 shows the adjacency matrix for undirected graph. A, B, C, D, E, and F represent the nodes. In the diagonal, all values are zero and if two nodes are connected, the matrix is denoted by the value of 1.

III. Approach

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & 0 & a_{23} & a_{24} & a_{25} \\ a_{12} & a_{13} & 0 & a_{14} & a_{15} \\ a_{12} & a_{13} & a_{14} & 0 & a_{15} \\ a_{12} & a_{13} & a_{14} & a_{15} & 0 \end{pmatrix} \end{array}$$

ID number	Adjacency matrix value
(1,1)	a_{11}
(1,2)	a_{12}
(1,3)	a_{13}
(1,4)	a_{14}
(1,5)	a_{15}
....	...
(5,4)	a_{54}
(5,5)	a_{55}

Figure 3: The resulting adjacency matrix including Id numbers in the first row and first column.

This outlier detection method is based on BEC for network data and p-value technique of hypothesis testing for finding outliers. For each data point, we calculate its BEC by using adjacency matrix for network data. To find out the adjacency matrix for the data set, we calculate the shortest paths through nodes in the destination IP. Figure 1 show the shortest path

through nodes in destination IP address. The numbers represent the label of each node for the given data points. The shortest path that is calculated creates an adjacency matrix from it by utilizing sparse matrices in order to increase computational speed. Our calculation is based on undirected network type data. The calculation for adjacency matrix yields an adjacency matrix from friendship nominations stored as a sparse matrix. The resulting adjacency matrix will include Id numbers in the first row and first column; it is shown in figure 3. To find the BEC, the calculation of the influence domain of each node in a given adjacency matrix for a given step, returns the undirected BEC for each node of undirected adjacency matrix 'adj'. Matrix 'adj' must be an undirected network and may or may not be sparse. The matrix is simple to change if the graph is directed.'adj' is assumed to have id numbers in the first row and also, this code could probably be more vectorised to speed up calculations for large adjacency matrices.

As our method needs to find the adjacency matrix for each data point, it is required to calculate the shortest path between each other data points. Since we have n data points, the complexity of calculating the shortest path is $O(n^2)$. Finally to find outliers we need to compare each data point with the other data points, thus resulting in $O(n^2)$ complexity.

IV. Experimental Results

This section describes the experiments and the results with synthetic data sets followed by how the data was generated. The experiment was run where τ was taken as 0.05. i.e., these experimental results have 95% confidence.

A. Synthetic Data

To cover the broad range of applications, network type data sets were generated. We apply a rigorous set of tests to the data in the path to understand the strength or weakness of the method. In all cases we use probabilistic distribution based data generation which takes user inputs to decide parameters of the data pattern. i.e., identify variables and then use a probabilistic model to generate the required number of data points and outliers.

After generating data, each set of data points with scaling features were tested by using both the BEC method and m-SNN [3] outlier detection method. The m-SNN method is a modification of the SNN (Shared Nearest Neighbor) method that aids in outlier detection.

In this analysis, we generated network data sets of three different sizes viz. small (100<), medium (100< medium< 1000) and large(1000>). An example for a small data set is a set with 56 total data points, where 6 of them were generated as global outliers which is small data set. After applying our new BEC method and m-SNN method with τ 0.05, all the expected global outliers were detected for the BEC method. Though the m-SNN approach was able to detect all the above labeled outliers correctly too, the results were not as accurate or precise as the BEC method.

The results obtained are summarized in Table II to demonstrate True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) values as percentages. It shows the average results for three different sizes of data sets. From the results, it is clear that the BEC has very high TP,TN percentages and very low FP,FN percentages compared to the m-SNN approach. Also the proposed method has the best results for the network type data. On comparing the results of complex path data sets, it is evident that the BEC is more robust in finding outliers (compared to m-SNN) particularly with respect to true positives and minimizing false negatives.

Procedure: Betweenness centrality Based Outlier Detection
Inputs: data[], a set of network data points;
Output: List of Outliers

// Finding Adjecency matrix for all the data points

Inputs: data[], Adjacency matrix for data points;
Output: List of Betweenness centrality for all data points

// Finding Betweenness centrality for all the data points

Inputs: data[], Betweenness Centrality for data points;
Output: List of Outliers
//Finding the outliers based on p-value method

Table 1: Betweenness centrality Based Outlier Detection Algorithm

	TP(%)	FP(%)	TN(%)	FN(%)
BEC	100.0	0.5	99.2	0.2
m-SNN	100.0	3.5	96.5	2.3

Table 2: Experimental results for BEC and m-SNN.

I. Conclusions

In this paper, we have described an algorithm based on graph theory capable of detecting outliers in different types of network type data sets. This method is a combination of adjacency matrix and betweeness centralities which avoids assumptions about data distributions and uses hypothesis testing to detect outliers. Through a series of experiments, we have shown that this method achieves good results with very high true positive and true negative values with the BEC approach producing outlier detection results equivalent or better than the m-SNN method. Furthermore, modifying this method can be used to identify an outlier to update a social network graph. Currently we are reformulating the algorithm to

improve the run time efficiencies and also to parallelize the code to make it amenable for massive data sets.

ACKNOWLEDGMENT

This research is funded in part by the Department of Energy under Contract Number DE-NA 0002686 and Clarkson Aerospace Corp. under subcontract CL-ATL-14-S7700-02-C2. Any opinions, findings, and conclusions or recommendations are those of the authors and expressed here those of the author(s) and do not necessarily reflect the views of the sponsors.

REFERENCE

[1] M. J. Lee, R. H. Choi, J.Lee, C. W. Chung and J. Y. Park, "QUBE: a Quick algorithm for Updating BEtweenness centrality," World Wide Web (WWW). ACM, pp. 351-360, 2012.

[2] D. Barbará, C. Domeniconi and J. P. Rogers, "Detecting outliers using transduction and statistical testing", ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), pp. 55-64, 2006.

[3] K. P. Liyanage, R. George and K. Shujaee, "Outlier Detection in Spatial Data using the m-SNN Algorithm", IEEE southeastCon 2013.

[4] J. P. Rogers "Detection of Outliers in Spatial-temporal Data", PhD Thesis.

[5] D. Hawkins, "Identification of Outliers", Chapman and Hall, London, 1980.

[6] T. Johnson, I. Kwok and R. Ng, "Fast Computation of 2-Dimensional Depth Contours", Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining, New York, NY, AAAI Press, 1998, pp. 224-228.

[7] E. M. Knorr and R. T. Ng, "Algorithms for Mining Distance-Based Outliers in Large Datasets", Proc. 24th Int. Conf. on Very Large Data Bases, New York, NY, 1998, pp. 293-298.

[8] E. M. Knorr and R. T. Ng, "Finding Intensional Knowledge of Distance-based Outliers", Proc. 25th Int. Conf. on Very Large Data Bases, Edinburgh, Scotland, 1999, pp. 211-222.

[9] E. Levent, M. Steinbach and V. Kumar, "A New Shared Nearest Neighbor Clustering Algorithm and its Applications".

[10] W. Wang, J. Yang and R. Muntz, "STING: A Statistical Information Grid Approach to Spatial Data Mining", Proc. 23th Int. Conf. on Very Large Data Bases, Athens, Greece, Morgan Kaufmann Publishers, San Francisco, CA, 1997, pp. 186-195.

[11] T. Zhang , R. Ramakrishnan and M. Linvy, "BIRCH: An Efficient Data Clustering Method for Very Large Databases", Proc. ACM SIGMOD Int. Conf. on Management of Data, ACM Press, New York, 1996, pp.103-114.

[12] F. Angiulli and C. Pizzuti, (2005) Outlier mining in large high-dimensional data sets. IEEE Transactions on Knowledge and Data Engineering, 17(2): 203-215.

[13] A. Gammerman and Vovk, V. (2002) Prediction algorithms and confidence measures based on algorithmic randomness theory. Theoretical Computer Science. 287: 209-217.

[14] UCI Machine Learning Repository. http://www.ics.uci.edu/ mlearn/MLRepository.html

[15] V. Vapnik, (1998) Statistical Learning Theory, New York: Wiley.

[16] M. Breunig, H. Kriegel, R. Ng and J. Sander, (2000) LOF: Identifying Density-Based Local Outliers. Proc. of the ACM SIGMOD Conference on Management of Data, 427- 438.

[17] K. Proedru, I. Nouretdinov, V. Vovk and A. Gammerman, (2002) Transductive confidence machine for pattern recognition. Proc. 13th European conference on Machine Learning. 2430:381-390.

[18] D. Barbara, C. Domeniconi and J. P. Rogers, "Detecting Outliers using Transduction and Statistical Testing", KDD'06, Philadelphia, Pennsylvania, 2006

[19] D. Velegrakis, "Outlier Detection over Data Streams using Statistical Modeling and Density Neighborhoods", Masters Thesis

[20] Eleazar Eskin. Anomaly detection over noisy data using learned probability distributions. Pages 255-262. Morgan Kaufmann, 2000.

[21] E. Eskin, W. Lee, and S. J. Stolfo, " Modeling system calls for intrusion detection with dynamic window sizes", In In proceedings of DARPA information Survivability Conference and Exposition 2 (DISCEX, 2001).

[22] M. Ester, H-P Kriegel, J. Sander and X. Xu, "A density based algorithm for discovering clusters in large spatial databases with noise", In Proc. Of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), pages 226-231, 1996.

[23] L. Ertoz, M. Steinbach and V. Kumar, "Finding topics in collections of documents: A shared nearest neighbor approach", In Workshop on Text Mining, held in conjunction with the First SIAM International Conference on Data Mining (SDM 2001). Society for Industrial and Applied Mathematics, 2003.

[24] S. Guha, R. Rastogi and K. Rock, "A robust clustering algorithm for categorical attributes," Inf. Syst., 25(5): 345-366, 2000.

[25] D. Barbara, N. Wu, and S. Jajodia, "Detecting novel network intrusions using bayes estimators," In Proceedings of the First SIAM Conference on Data Mining, April 2001.

[26] R. Bronstein, J. Das, M. Duro, R. Friedrich, G. Kleyner, M. Muller, S. Singhal and I. Cohen, "Self-aware services: Using Bayesian networks for detecting anomalies in internet-based services," In Northwestern University and Stanford University Gary Igor, Pages 623-638, 2001.

[27] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," The VLDB Journal, 8(3-4): 237-253, 2000.

[28] M. Markou and S. Singh, "Novelty detection," A review – part 1: Statistical approaches. Signal Processing, 83: 2003, 2003.

[29] J. C. Maxwell, "A Treatise on Electricity and Magnetism," 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[30] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[31] K. Elissa, "Title of paper if known," unpublished.

[32] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[33] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[34] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[35] J. Anthonisse and S. M. C. A. A. M. besliskunde, "Therush in a directed graph", Technical report, 1971.

[36] D. A. Bader, S. Kintali, K. Madduri, and M. Mihail, "Approximating betweenness centrality", In Proceedings of the 5th international conference on Algorithms and models for the web-graph, WAW'07, pages 124–137, Berlin, Heidelberg, 2007. Springer-Verlag.

[37] U. Brandes, "A faster algorithm for betweenness centrality". Journal of Mathematical Sociology, 25(1994):163–177, 2001.

[38] U. Brandes, "On variants of shortest-path betweenness centrality and their generic computation", Social Networks, 30(2):136–145, 2008.

[39] U. Brandes and C. Pich, "Centrality estimation in large networks" International Journal Of Bifurcation And Chaos, 17(7):2303, 2007.

[40] L. C. Freeman, "A set of measures of centrality based on betweenness", Sociometry, 40(1):35–41, 1977.

[41] R. Geisberger, P. Sanders, and D. Schultes, "Better approximation of betweenness centrality", In J. I. Munro and D. Wagner, editors, ALENEX, pages 90–100. SIAM, 2008

[42] . M. E. J. Newman. "A measure of betweenness centrality based on random walks", Social Networks, 27(1):39–54, 2005.

[43] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks", Physical Review E, 69(2): 26113, 2004.

[44] J. W. Pinney and D. R. Westhead, "Betweenness-based decomposition methods for social and biological networks", In Interdisciplinary Statistics and Bioinformatics, pages 87–90. Leeds University Press, 2006.

A Generalization of the PageRank Algorithm

Z. Bahrami Bidoni, R. George, K.A. Shujaee
Department of Computer and Information Systems
Clark Atlanta University
Atlanta, GA
{zeynab.bahrami, rgeorge, kshujaee}@cau.edu

Abstract— **PageRank is a well-known algorithm that has been used to understand the structure of the Web. In its classical formulation the algorithm considers only forward looking paths in its analysis- a typical web scenario. We propose a generalization of the PageRank algorithm based on both out-links and in-links. This generalization enables the elimination network anomalies- and increases the applicability of the algorithm to an array of new applications in networked data. Through experimental results we illustrate that the proposed generalized PageRank minimizes the effect of network anomalies, and results in more realistic representation of the network.**

Keywords- Search Engine; PageRank; Web Structure; Web Mining; Spider-Trap; dead-end; Taxation;Web spamming.

I. INTRODUCTION

With the rapid growth of the Web, users can get easily lost in the massive, dynamic and mostly unstructured network topology. Finding users' needs and providing useful information are the primary goals of website owners. Web structure mining [1],[2],[3] is an approach used to categorize users and pages. It does so by analyzing the users' patterns of behavior, the content of the pages, and the order of the Uniform Resource Locator (URL) that tend to be accessed. In particular, Web structure mining plays an important role in guiding the users through the maze. The pages and hyperlinks of the World-Wide Web may be viewed as nodes and arcs in a directed graph. The problem is that this graph is massive, with more than a trillion nodes, several billion links, and growing exponentially with time. A classical approach used to characterize the structure of the Web graph through PageRank algorithm, which is the method of finding page importance.

The original PageRank algorithm [3],[4],[5] one of the most widely used structuring algorithms, states that a page has a high rank if the sum of the ranks of its backlinks is high. Google effectively applied the PageRank algorithm, to the Google search engine [4]. Xing and Ghorbani [6] enhanced the basic algorithm through a Weighted PageRank (WPR) algorithm, which assigns a larger rank values to the more important pages rather than dividing the rank value of a page evenly among its outgoing linked pages. Each outgoing link page gets a value proportional to its popularity (its number of in-links and out-links). Kleinberg [7] identifies two different forms of Web pages called hubs and authorities, which lead to the definition of an iterative algorithm called Hyperlink Induced Topic Search (HITS) [8].

Bidoki and Yazdani [9] proposed a novel recursive method based on reinforcement learning [10] that considers distance between pages as punishment, called "DistanceRank" to compute ranks of web pages in which the algorithm is less sensitive to the "rich-get-richer" problem [9],[11] and finds important pages faster than others. The DirichletRank algorithm has been proposed by X. Wang et al [12] to eliminate the zero-one gap problem found in the PageRank algorithm proposed by Brin and Page [4]. The zero-one gap problem occurs due to the ad hoc way of computing transition probabilities. They have also proved that this algorithm is more robust against several common link spams and is more stable under link perturbations. Singh and Kumar [13] provide a review and comparison of important PageRank based algorithms.

As search engines are used to find the way around the Web, there is an opportunity to fool search engines into leading people to particular page. This is the problem of web spamming [14], which is a method to maliciously induce bias to search engines so that certain target pages will be ranked much higher than they deserve. This leads to poor quality of search results and in turn reduces the trust in the search engine. Consequently, anti-spamming is a big challenge for all the search engines. Earlier Web spamming was done by adding a variety of query keywords on page contents regardless of their relevance. In link spamming [15], the spammers intentionally set up link structures, involving a lot of interconnected pages to boost the PageRank scores of a small number of target pages. This link spamming does not only increasing the rank gains, but also makes it harder to detect by the search engines. It is important to point out that link spamming is a special case of the spider-traps [16]. At the present time, the Taxation method [16] is the most significant way to diminish the influence of the spider-traps and dead-ends by teleporting the random surfer to a random page in each iteration.

This article has two main contributions: First, we present a generalized formulation of the PageRank algorithm based on transition probabilities, which takes both in-link and out-links of node and their influence rates into account in order to calculate PageRanks. This would permit the application of this approach to a wide variety of network problems that require consideration of the current state values (and PageRank) as a function of past state transitions. Second, we describe a novel approach of adding virtual edges to a graph that permits more realistic computations of PageRank,

 ISBN: 978-1-61208-324-7

negating the effect of network anomalies such as spider-traps and dead-ends.

The paper is organized as follows. In Section 2, a brief background review of the basic concepts for computing PageRanks based on transition probabilities is presented and the problems related to network anomalies such as spider-traps and dead-ends together with their solution method based on Taxation is stated. In Section 3, we introduce the proposed general approach for determining PageRank. In Section 4, we apply our PageRank method to a typical graph with all types of possible structures and inter/ intra-correlations and compare our results with the baseline technique. In Section 5, we conclude by describing the contribution of our method and discuss its results.

II. OVERVIEW ON THE PAGERANK APPROACH BASED ON TRANSITION PROBABILITIES

PageRank is a function that assigns a real number to each page in the Web. We begin by defining the basic, idealized PageRank, and follow it by modifications that are necessary for dealing with some real-world problems concerning the structure of the Web. Imagine surfing the Web, going from page to page by randomly (random surfer) choosing an outgoing link from one page to get to the next. This can lead to dead-ends at pages with no outgoing links, or cycles around cliques of interconnected pages. This theoretical random walk is known as a Markov chain or Markov process [16],[17].

In general, we can define the transition matrix of the Web to describe what happens to random surfers after one step. This matrix M has n rows and columns, if there are n pages. The element m_{ij} in row i and column j has value $1/k$ if page j has k arcs out, and one of them is to page i. Otherwise, $m_{ij} = 0$. The probability distribution for the location of a random surfer can be described by a column vector whose jth component is the probability that the surfer is at page j. This probability is the (idealized) PageRank function.

Suppose we start a random surfer at any of the n pages of the Web with equal probability. Then the initial vector v_0 will have $1/n$ for each component. If M is the transition matrix of the Web, then after one step, the probability distribution of the surfer place will be Mv_0, after two steps it will become $M(Mv_0) = M^2 v_0$, and so on. In general, multiplying the initial vector v_0 by M a total of i times will give us the distribution of the surfer after i steps.

This sort of behavior is an example of a Markov processes. It is known that the distribution of the surfer approaches a limiting distribution v that satisfies $v = Mv$, provided two conditions are met:

1) The graph is strongly connected; that is, it is possible to get from any node to any other node.

2) There are no dead-ends: nodes that have no arcs out.

In fact, because M is stochastic, meaning that each of its columns adds up to 1, v is the principal eigenvector. Note also that, because M is stochastic, the eigenvalue associated with the principal eigenvector is 1.The principal eigenvector of M tells us where the surfer is most likely to be after infinite steps i. The intuition behind PageRank is that the more likely a surfer is to be at a page, the more important the page is. We can compute the principal eigenvector of M by starting with the initial vector v_0 and multiplying by M some number of times, until the vector we get shows little change at each round. In practice, for the Web itself, 50–75 iterations are sufficient to converge to within the error limits of double-precision arithmetic.

A. Structure of the Web

It would be nice if Web pages were strongly connected. However, it is not the case in practice. An early study of the Web found it to have the structure shown in Figure 1. There is a large strongly connected component (SCC), but there were several other portions that were almost as large [18].

- The **in-component**, consisting of pages that could reach the SCC by following links, but were not reachable from the SCC.
- The **out-component**, consisting of pages reachable from the SCC but unable to reach the SCC.
- **Tendrils**, which are of two types. Some tendrils consist of pages reachable from the in-component but not able to reach the in-component. The other tendrils can reach the out-component, but are not reachable from the out-component.

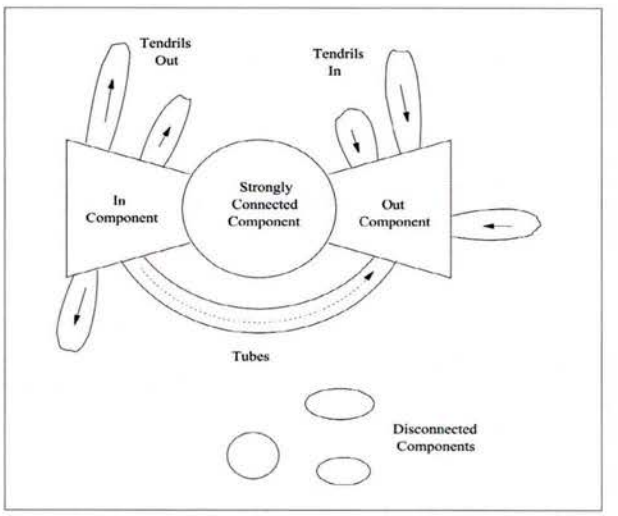


Figure 1. The "bowtie" representation of the Web [22]

In addition, there were small numbers of pages found either in

- Tubes, which are pages reachable from the in-component and able to reach the out-component, but unable to reach the SCC or be reached from the SCC.
- Isolated components that are unreachable from the large components (the SCC, in- and out-components) and unable to reach those components.

 ISBN: 978-1-61208-324-7

As a result, PageRank is usually modified to prevent such anomalies. There are, in principle, two problems we need to avoid. First, is the dead-end - a page that has no links out- which will bring a zero column in the forward transition matrix, and consequently it will cause all PageRanks to become zero. The second problem is groups of pages that all have out-links but they never link to any other pages. These structures are called spider-traps. Both these problems are solved by a method called "taxation," where we assume a random surfer has a finite probability of leaving the Web at any step, and new surfers are started at each page.

B. Taxation

To avoid the problem of spider-trap or dead-end, we modify the calculation of PageRank by allowing each random surfer a small probability of teleporting to a random page, rather than following an out-link from their current page. The iterative step, where we compute a new vector estimate of PageRanks v' from the current PageRank estimate v and the transition matrix M is

$$v' = \beta M v + (1-\beta)e / n \tag{1}$$

Where β is a chosen constant, usually in the range 0.8 to 0.9, e is a vector of all 1's with the appropriate number of components, and n is the number of nodes in the Web graph. The term βMv represents the case where, with probability β, the random surfer decides to follow an out-link from their present page. The term $(1-\beta)e/n$ is a vector each of whose components has value $(1-\beta)/n$ and represents the introduction, with probability $1-\beta$, of a new random surfer at a random page.

Although by employing this formulation, the effect of spider-trap and dead-end is controlled and the PageRank is distributed to each of other nodes, components of spider-trap still are managed to get most of the PageRank for themselves. Therefore, the PageRanks of nodes are still unreasonable. For instance, in Figure 2. , C is a simple spider trap of one node and the transition matrix is as follows:

$$M = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \tag{2}$$

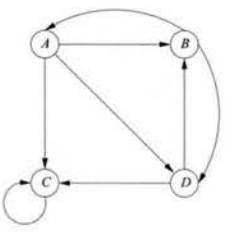


Figure 2. A graph with a one-node spider trap

If we perform the usual iteration to compute the PageRank of the nodes, we get

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \begin{bmatrix} 3/24 \\ 5/24 \\ 11/24 \\ 5/24 \end{bmatrix} \begin{bmatrix} 5/48 \\ 7/48 \\ 29/48 \\ 7/48 \end{bmatrix} \begin{bmatrix} 21/288 \\ 31/288 \\ 205/288 \\ 31/288 \end{bmatrix} \cdots \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{3}$$

As predicted, all the PageRank is at C, since once there a random surfer can never leave. To avoid the problem illustrated, we modify the calculation of PageRank by the Taxation method. Thus, the equation for the iteration becomes

$$\mathbf{v}' = \begin{bmatrix} 0 & 2/5 & 0 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 4/5 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{bmatrix} \mathbf{v} + \begin{bmatrix} 1/20 \\ 1/20 \\ 1/20 \\ 1/20 \end{bmatrix} \tag{4}$$

Notice that we have incorporated the factor β into M by multiplying each of its elements by *4/5*. The components of the vector $(1-\beta)e/n$ are each *1/20*, since $1-\beta = 1/5$ and $n=4$. The first iteration:

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \begin{bmatrix} 9/60 \\ 13/60 \\ 25/60 \\ 13/60 \end{bmatrix} \begin{bmatrix} 41/300 \\ 53/300 \\ 153/300 \\ 53/300 \end{bmatrix} \begin{bmatrix} 543/4500 \\ 707/4500 \\ 2543/4500 \\ 707/4500 \end{bmatrix} \cdots \begin{bmatrix} 15/148 \\ 19/148 \\ 95/148 \\ 19/148 \end{bmatrix} \tag{5}$$

By being a spider trap, C has still managed to get more than half of the PageRank for itself. However, the effect has been limited, and each of the nodes gets some of the PageRank.

III. A Generalized Method

In web arena, a link by important pages will impact on significance of a page. However, there are other networks in which not just in-link but out-links are also weighty. For instance, in social networks, connecting to eminent people (out-link) is as crucial as being connected by key persons (in-link) in evaluating the degree of prominence of a member. Therefore, sometimes sorting and grading nodes of a graph only based on in-links will result in an incorrect evaluation. So, we take out-links and the rate of their impacts with respect to in-links into our computations.

A. Algorithm

Suppose we start as a random surfer at any of the n pages of the Web with equal probability. Then the initial vector will have $1/n$ for each component. If M_f is the forward transition matrix of the Web, then after one forward step, the probability distribution of the next surfer place will be $M_f v_0$ and if M_b is the backward transition matrix of the Web, then after one backward step, the probability distribution of the previous surfer place will became $M_b v_0$. Also, we consider the importance weight factor of both in-links (β) and out-links ($1-\beta$).

Note that equation $\left(\beta M_f + (1-\beta) M_b\right)$ is the linear combination of both next and previous surfer place, and it is

also stochastic because it is a linear combination of two stochastic matrices. So its eigenvalue associated with the principal eigenvector will be 1. The principal eigenvector of $(\beta M_f + (1-\beta)M_b)$ tells us where the surfer is most likely to be after a long time. Recall that the intuition behind PageRank is that the more likely a surfer is to be at a page, the more important the page is. We can compute the principal eigenvector of $(\beta M_f + (1-\beta)M_b)$ by starting with the initial vector v_0 and multiplying by $(\beta M_f + (1-\beta)M_b)$ some number of times, until the vector we get shows little change at each round. Considering this matrix instead of M_f has two advantages: First, in computing PageRank of a node, the importance of its neighbors with both types of relationship (out-link and in-link) and their arbitrary impact rates (parameter β) have taken into account. Second, by using this method, we do not have the problems about dead-ends and spider-traps because we take the linear combination of entering probability from and exiting probability to other nodes in our computation. Therefore, in case $\beta \neq 0$ and $\beta \neq 1$, the columns related to dead-ends are not completely zero. Likewise, for the spider-trap columns, probabilities related to other nodes are not zero and they cannot absorb more unreasonable rank to themselves. About cases $\beta = 1$ or $\beta = 0$, in the following, we proposed another idea (adding virtual edges) by which the random surfer can exit from dead-ends and spider-traps.

The proposed algorithm is as follows:

Step 1: finding Forward and Backward transition matrices.

Step 2: considering appropriate formula and keep iterating until it gets converged.

In this step, three possible conditions can exist which are characterized as following:

Case 1: $\beta \neq 0$ **and** $\beta \neq 1$. It means that both forward and backward trends are important to calculate PageRanks. Thus, we only need to calculate the eigenvector of matrix $(\beta M_f + (1-\beta)M_b)$.

Case 2: $\beta = 1$ So, we need only the forward matrix to calculate PageRanks. If there are not a dead-end or a spider-trap in the graph, the vector of PageRanks is the eigenvector of M_f. If there are dead-ends or spider-traps, the eigenvector of M_f assigns most of PageRank to spider-traps and dead-ends that is not real. Thus we add enough virtual out-links to remove these spider and dead-end situations. For each dead-end and spider-trap, we will consider a virtual edge in which source of them are dead-ends and one member of each spider-traps, respectively. Also, their destinations can be any arbitrary nodes, excepting those of dead-end and spider-traps (see Figure 3. Green color edges). Hence, If assumed v is eigenvector of matrix M_f' (forward transition matrix after adding virtual links), in order to find final PageRanks of vertices, we have to remove effect of these virtual links on PageRanks by calculating the following equation $v - (M_f' - M_f)v$.

Case 3: $\beta = 0$. Here only backward trend (out-links) is important to consider for calculation of PageRanks. So we only need backward matrix to determine PageRanks. If there are not in-component or in-tendril vertices in the graph, vector of PageRanks is eigenvector of M_b. If there are in-component or in-tendril vertices, eigenvector of M_b assigns most of PageRank to in-component and in-tendril vertices, which is not real. Thus we add enough virtual in-links to remove these in-component and in-tendril situations then after computing eigenvector of new backward matrix M_b', we have to remove effect of these virtual links on PageRanks (see Figure 3. Red color edges). If suppose v is eigenvector of matrix M_b' (backward transition matrix after adding virtual links). The final PageRanks of vertices would be $v - (M_b' - M_b)v$.

Step 3: normalize PageRank vector to find distribution probability of vertices.

As shown below, if we consider a matrix include the importance of pairwise comparison of vertices (A), eigenvector of this matrix would be distribution probability of vertices.

Note that, W is vector distribution probability of vertices that sum of its components is 1 and also w_i is amount of vertex i's importance. So, instead of w_i / w_j in matrix A, we let p_i / p_j, which p_i, p_j are PageRanks of nodes i, j. We calculate eigenvector of matrix A and to get the distribution probability of vertices.

$$AW = \begin{bmatrix} w_1/w_1 & w_1/w_2 & \cdots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & \cdots & w_2/w_n \\ \vdots & \vdots & \cdots & \vdots \\ w_n/w_1 & w_n/w_2 & \cdots & w_n/w_n \end{bmatrix} * \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = n * \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = nW \quad (6)$$

B. Biased Random Walk

In order to bias the rank of all nodes with respect to a special subset of nodes, we use the Biased Random Walk method in which the random surfer, in each iteration, will jump on one of the member of the subset with equal probability. Its most important application is topic-sensitive PageRank [19] in search engines. The consequence of this approach is that random surfers are likely to be at an identified page, or a page reachable along a short path from

 ISBN: 978-1-61208-324-7

one of these known pages, because the pages they link to are also likely to be about the same topic. The mathematical formulation for the iteration that yields topic-sensitive PageRank is similar to the equation we used for general PageRank. The only difference is how we add the new surfers. Suppose S is a set of integers consisting of the row/column numbers for the pages we have identified as belonging to a certain topic (called the teleport set). Let e_s be a vector that has 1 in the components in S and 0 in other components. Then the topic-sensitive PageRank for S is the limit of the iteration

$$v' = \alpha(\beta M_f + (1-\beta)M_b)v + (1-\alpha)e_s / |s| \quad 0.8 \le \alpha \le o.9 \tag{7}$$

Here, as usual, M is the transition matrix of the Web, and $|S|$ is the size of set S.

IV. THE EXPERIMENT

Figure 3. is a graph with 20 vertices that include all kinds of network artifacts mentioned in section 2.

SCC:{1,2,4,5,7,8,9,10,15,17,18,20} TUBE:{16-6}

OUT-COMPONENT:{6,11,12} IN-COMPONENT:{3,13,16}
OUT-TENDRIL:{14} IN-TENDRIL:{19}

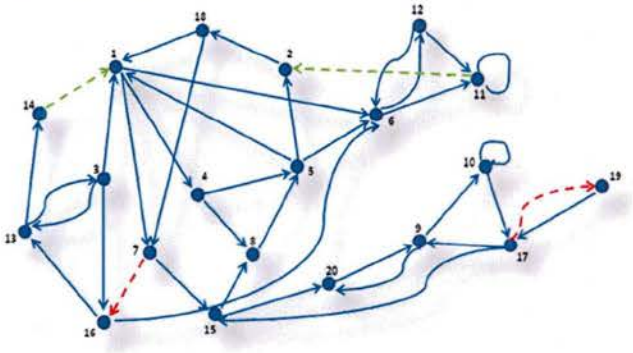


Figure 3. Synthetic Graph Example

In case 2 ($\beta = 1$), there are a dead-end situation on vertex 14 and a spider-trap situation on set of vertices {6, 11, 12}, and in order to remove the dead-end and the spider-trap consider 2 virtual out-link (green edges) on these vertices. Also in case 3 ($\beta = 0$), there are in-component situation on set of vertices {3, 13, 16}, and in order to remove negative PageRank consider 2 virtual in-link (red edges) on these vertices. For completeness, we also compute the biased random walk on case1. Comparing the results with case1, TABLE I. , it is clear that PageRanks are biased on set S={2, 4, 7, 18}. As we expect, rank of nodes of set S and nodes that are pointed by set S get higher ranks.

TABLE I. PAGERANK VECTOR AT CASES 1, 3, AND BIASED RANDOM WALK.

Results of case 1 ($\beta = 0.7$)		Results of the biased random walk on case1		Results of case 3 ($\beta = 0$)	
Nodes number	*PageRank*	*Nodes number*	*PageRank*	*Nodes number*	*PageRank*
11	0.945	5	0.9937	17	0.57916
12	0.2177	11	0.9878	10	0.38611
6	0.1767	18	0.9703	13	0.36037
9	0.0703	1	0.9432	1	0.27028
10	0.0632	7	0.9013	3	0.27028
5	0.0601	15	0.8513	5	0.25741
1	0.0543	2	0.7444	9	0.25741
20	0.0527	4	0.6847	7	0.24454
15	0.0495	6	0.65	4	0.19305
17	0.045	8	0.6414	19	0.19305
8	0.036	9	0.5045	16	0.18018
7	0.029	20	0.4878	2	0.16731
4	0.0272	12	0.3659	18	0.16731
18	0.025	10	0.3204	8	0.1287
3	0.0237	17	0.2976	15	0.1287
13	0.023	3	0.1628	20	0.1287
16	0.0223	13	0.1144	12	1.14E-17
2	0.0216	16	0.0923	6	7.34E-18
14	0.0081	19	0.0386	11	0
19	0.0068	14	0.035	14	0

TABLE II. COMPARING RESULTS OF THE ALGORITHM AND TAXATION METHOD TO AVOID ANOMALIES IN CASE 2 ($\beta = 1$)

Using virtual edges		Taxation	
nodes no	*PageRank*	*nodes no*	*PageRank*
9	0.508068237	11	0.83086
10	0.508068237	9	0.25352
20	0.381051178	10	0.22903
2	0.265581124	20	0.19944
17	0.254034118	15	0.15968
15	0.254034118	6	0.1495
5	0.173205081	5	0.14569
18	0.161658075	17	0.14155
8	0.15011107	8	0.11547
1	0.138564065	1	0.11197
6	0.138564065	7	0.08907
7	0.127017059	12	0.08748
11	0.103923048	18	0.07921
12	0.069282032	2	0.06521
4	0.046188022	4	0.05567
3	7.50E-17	13	0.0528
13	2.12E-17	3	0.04612
16	1.16E-17	14	0.04612
14	1.02E-17	16	0.0369
19	0	19	0.02386

Comparing the results of the Taxation method and our proposed method, TABLE II. , obviously we can realize that our approach produces more reasonable outcomes. Because, as it is shown in the TABLE II, node 9 is the junction of two cycles, all nodes of these cycles are from SCC part of the graph, so the random surfer is most likely on it. The nodes 10 and 20 have higher rank after 9, because they have in-link from the node 9. The rank of node 5 cannot be higher than 17 because the node 17 is a member of the cycle consist of

 ISBN: 978-1-61208-324-7

node 9 and 10. In Taxation result, the nodes with spider-trap situation such as 6 and 11 got higher and vertices 2 and 18 got lower PageRank than our proposed approach results. Also, for other vertices, their ranks are either the same or very close to each other's.

V. CONCLUSION

In this paper, the fundamental idea of Web Structure mining and Web Graph is explained in detail to have a generic understanding of the data structure used in web. The main purpose of this paper is to present the new PageRank based algorithms and compare that with the previous algorithms.

The proposed method generalizes the approach of finding PageRank based on transition probabilities by considering the arbitrary impact rates of both out-links and in-links, in order to include all possible cases because there are some conditions in which out-links have also an influence on PageRank of nodes. Moreover, it prevents that spider-traps and dead-ends have a high unreasonable rank and assign higher PageRanks to themselves. The noticeable weak point of previous method is that it assigns more unreasonable PageRank to spider-traps and dead-ends, and also reduces PageRank of SCC vertices. But in our approach this problem has been solved, because by adding virtual edges, random surfers will not stop on spider-traps and dead-ends. According to [13], DirichletRank has been so far the best method amongst previous methods, capable of diminishing the impact of link spamming (a special case of spider-traps) and dead-end problem that is, however, only applicable to backward analysis. Our approach in comparison with their method is general for more types of networks and simpler to understand and implement. Also, by using ideas suggested in this paper, in any possible cases, PageRanks is insulated from the influence of anomalies including in/out-tendrils and in/out-components.

The generalization of the PageRank algorithm to include forward and backward links into a node makes this approach applicable to new domains beyond web mining and search engines. We are currently exploring the application of the new generalized algorithm to the analysis of network data for instance using PageRank as a measurement of node's activity score [20] to find communities.

ACKNOWLEDGMENT

This research is funded in part by the Army Research Laboratory under Grant No: W911NF-12-2-0067 and Army Research Office under Grant Number W911NF-11-1-0168. Any opinions, findings, conclusions or recommendations expressed here are those of the author(s) and do not necessarily reflect the views of the sponsor.

REFERENCES

[1] R. Kosala and H. Blockeel, "Web mining research: A survey," ACM SIGKDD Explorations, 2(1), 2000, pp. 1–15.

[2] S. Madria, S. S. Bhowmick, W. K. Ng, and E.-P. Lim, "Research issues in web data mining," In Proceedings of the Conference on Data Warehousing and Knowledge Discovery, 1999, pp. 303–319.

[3] S. Pal, V. Talwar, and P. Mitra, "Web mining in soft computing framework : Relevance, state of the art and future directions," IEEE Trans. Neural Networks, 13(5), 2002, pp. 1163–1177.

[4] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Technical report,Stanford Digital Libraries SIDL-WP-1999-0120, 1999.

[5] C. Ridings and M. Shishigin, "Pagerank uncovered," Technical report, 2002.

[6] W. Xing and A. Ghorbani, "Weighted PageRank Algorithm," Proc. of the Second Annual Conference on Communication Networks and Services Research (CNSR '04) IEEE, 2004 , pp. 305-314, 0-7695-2096-0/04 .

[7] J. Kleinberg, "Authoritative Sources in a Hyper-Linked Environment", Journal of the ACM 46(5), 1999, pp. 604-632.

[8] S. Chakrabarti, et al. "Mining the Web's link structure." Computer 32.8 ,1999, pp. 60-67.

[9] A. M. Zareh Bidoki and N. Yazdani, "DistanceRank: An intelligent ranking algorithm for web pages," Information Processing and Management, Vol 44, No. 2, 2008, pp. 877-892.

[10] R.S. Sutton and A.G. Barto, "Reinforcement Learning: An Introduction," Cambridge, MA: MIT Press, 1998.

[11] J. Cho, S. Roy and R. E. Adams, "Page Quality: In search of an unbiased web ranking," Proc. of ACM International Conference on Management of Data,". 2005, pp. 551-562.

[12] X. Wang, T. Tao, J. T. Sun, A. Shakery, and C. Zhai, "DirichletRank: Solving the Zero-One Gap Problem of PageRank," ACM Transaction on Information Systems, Vol. 26, Issue 2, 2008.

[13] A. K. Singh and P. Ravi Kumar. "A Comparative Study of Page Ranking Algorithms for Information Retrieval," International Journal of Electrical and Computer Engineering 4, no. 7 (2009), pp. 469-480.

[14] Z. Gyongyi and H. Garcia-Molina, "Web Spam Taxonomy," First International Workshop on Adversarial Information Retrieval on the Web *(AIRWeb 2005)*, 2005.

[15] Z..Gyongyi and H. Garcia-Molina, "Link Spam Alliances," Proc. of the 31st International Conference on Very Large DataBases (VLDB), 2005, pp. 517-528.

[16] A. Rajaraman, J. Leskovec, and J. D. Ullman, "Mining of Massive Datasets," 2013, pp.161-198.

[17] S. Brin and L. Page, "Anatomy of a large-scale hypertextual web search engine," Proc. 7th Intl. World-Wide-Web Conference, 1998, pp. 107–117.

[18] A. Broder, et al. "Graph structure in the web," *Computer networks* 33.1 , 2000, pp. 309-320.

[19] T.H. Haveliwala, "Topic-sensitive PageRank," Proc. 11th Intl. World-Wide-Web Conference, 2002, pp. 517–526.

[20] J. Qiu and Z. Lin, "A framework for exploring organizational structure in dynamic social networks," Decision Support Systems, 51, 2011, pp.760–771.

+

 ISBN: 978-1-61208-324-7

Discovering Community Structure in Dynamic Social Networks using the Correlation Density Rank

Zeynab Bahrami Bidoni [1], Roy George[2]
Department of Computer and Information Systems
Clark Atlanta University, Atlanta, GA
zeynab.bahramibidoni@students.cau.edu [1]; rgeorge@cau.edu [2]

Abstract

Recent research has produced advances in the understanding of communities within a dynamic social network. A "community" in this context is defined as a subgraph with a higher internal density and a lower crossing density with respect to other subgraphs. In this paper, we describe a novel and efficient distance based ranking algorithm, called the "Correlation Density Rank" (CDR), which is utilized to derive the community tree from the social network and to develop a tree learning algorithm that is employed to construct an evolving community tree. Also, we present an evolution graph of the organizational structure, through which new insights into the dynamic network may be obtained. The experiments, conducted on a datasets, both synthetic and real, demonstrate the feasibility and applicability of the framework.

Keywords: Dynamic social network; Organizational structure; Community discovery; Evolution analysis; Web ranking; Crawling; Correlation Density Rank.

1. Introduction

Community detection is an important research issue in social network analysis (SNA), where the objective is to recognize related sets of members such that intra-community associations are denser than inter-communities associations [1-10]. Researchers have presented various methods to extract communities from a Social Network (SN) data. . In particular, discovering the organizational structure of communities in an SN has been identified as an interesting but challenging problem [11,12] Examples of important applications include characterizing potential candidates for viral marketing, finding members of criminal groups, discovering affinity groups, etc. [12] While there has been research on finding key members in an SN [11-15] the results have limited power to supply a complete view of the organizational structure.

In the real world, social networks are constantly changing and evolving. New members may join the network, existing members may quit from the network, and associations among members constantly change over time. Therefore, the approach should be capable of supporting the exploration of organizational structure dynamically. Some earlier research has provided approaches to detecting communities from a dynamic social network [16-19] These approaches discover changes of communities in an SN, but do not answer questions related to changing organizational structure, such as who is becoming more powerful or the shifting of the power structure.

In the workflow area, organization mining has also been the focus of past research [20-22]. The workflow area research has emphasized the exploration of organizational structure in the unit from event logs of information systems. Research efforts [23-25] have also addressed determining the hierarchy in an SN, where hierarchy has a similarity to an organizational structure. In this paper we use the notion of a community tree data structure to represent organizational structure and its evolution. This approach is similar to that presented by Qui and Lin [26] where the ranking of nodes is based on PageRank algorithm with iteration complexity of the order of logN [27]. However, the technique presented here has no iteration complexity and is not susceptible to network anomalies [28] such as spider-traps, dead-ends, etc. and the "rich-get-richer" problem [29]. Consequently, the approach in this paper is algorithmically efficient and produces communities and organizational structures with better accuracy.

We propose a novel density measure, the Correlation Density Rank (CDR), as the basis of community and organizational structure detection. We apply a density based method to describe the relationship between nodes linked by edges. In comparison with other earlier Density based algorithms [30-32], this approach offers several advantages (1) the CDR is a generalized formulation, which permits the weighting of different correlation types, . (2) a more realistic solution where with important neighbors have more priority than lesser ones is employed, and (3) optimization of the CDR is easier, since the number of parameters needed for tuning is smaller.

The contributions of this paper are as follows: (1) Developing an algorithm to derive the community tree from the SN. (2) Developing a tree learning algorithm to generate the evolving community trees. (3) Proposing an approach for representing the evolution of the organizational structure based on the evolving community trees.

The rest of the paper is organized as follows. Section 2 introduces the concept of community tree and proposes an approach for deriving the community tree from a static social network. Section 3 presents the methods for analyzing the evolution of organizational structure in a dynamic SN. Section 4 provides the experimental results. Section 5 offers concluding remarks.

ISBN: 978-1-62561-000-3

2. Discovering the organizational structure in a static social network

2.1 Organization structure of a network

Song and Van [33] developed a definition of organizational model, where the organizational model consists of organizational units (e.g. functional units), roles (e.g. duty), originators, and relationships (e.g., hierarchy). In this paper, we assume that the organizational structure of an SN is a hierarchy that represents communities (or units) and subordinations of members in the SN. However, the derived hierarchical organizational structure of an SN is not necessarily reflected in a real-world organization [26]. For instance, there exists real organizations that are cannot be characterized as an academic network, nor in a blogosphere. In this case for a network formed by e-mail communications, we may not be able to exactly determine to what extent the relationships can be mapped to the real-world organizational structure. Therefore, in deriving the organizational relationships, subordination describes the relationship between two members where the leader is the most likely and more important destination of information flow starting from the subordinate. i.e., the subordinate has a closest interaction with the leader in comparison with others while the leader has a higher possibility of attracting interaction with the nodes across the entire network in comparison to the subordinate nodes.

The importance of members in the SN, is depicted through a score, the m-Score, which is equivalent to CDR value of a member. The m-Score value of a member would be its certainty on attracting interaction with all nodes through whole network. The higher the score is, the more important the member is. We use a data structure, called the community tree, to represent the SN organizational structure.

Definition 1. (Community Tree): Let $N = \{n_1,...,n_k\}$ be a collection of members in an SN, CT is a tree, and NULL is the root of the tree, and every member in N is referred to as a node in the tree. Each member n_i in CT it has a unique parent node n_j where m-Score (n_j) > m-Score (n_i). If the parent node of n_i is the root node NULL, n_i is called a core of the tree. A core and its descendants compose a community.

To derive the organizational structure, we calculate the m-Score for every member and then attempt to find the immediate leader of every member in a network. Further, we construct the community tree using m-Score and subordinations.

After having constructed the SN community tree, we can discover communities and obtain the SN organizational structure. An example of a community tree is illustrated in Fig. 1 where node 1 and node 5 are cores of community 1 and community 2, respectively.

This research is funded in part by the Army Research Laboratory under Grant No: W911NF-12-2-0067 and Army Research Office under Grant Number W911NF-11-1-0168. Any opinions, findings, conclusions or recommendations expressed here are those of the author(s) and do not necessarily reflect the views of the sponsor.

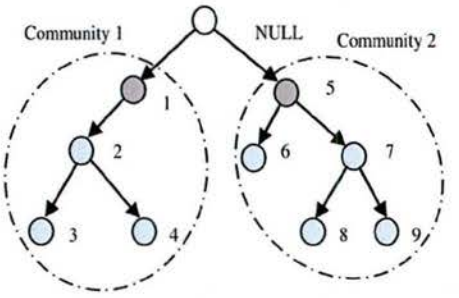


Figure 1. An example of a community tree [26].

SN may be regarded as a graph indicating information flows among members. The relationship of SN members can be obtained by analyzing information flows [34]. The process of determining SN information flow is similar to a random walk on a graph [35,36]. Given a graph and a starting point, the starting point's neighbor is selected at random, and the next start point is moved to this neighbor then a neighbor of this new start point is selected again at random and so on. The random sequence of points generated in this process is a random walk on the graph. The expected lengths of random walks on the graph, can be used to derive randomized shortest paths (RSP) dissimilarity [37,38]. The RSP dissimilarity, which has its foundation in statistical physics, may be used to compute the shortest path distance for all pairs of nodes of a graph in closed form. Recently [39] has generalized distance from the RSP framework based on the Helmholtz free energy between two states of a thermodynamic system with a distance measure, the free energy distance. We employ the RSP measurement method in [39] as the distance between nodes, but with one major difference: we consider customized initial cost for edges such that, along with finding shortest path between nodes. The random walker intelligently selects the most important neighbor resulting in lower cost and smaller distance.

Combining RSP with m-Score of every node, we can find an immediate leader for every member in an SN (see section 2.3). The SN community tree is derived in this way. Our framework includes the following steps to derive the community tree:

a) *Employ the novel "Correlation Density Rank" method to ranking nodes as the m-Score for every node in an SN; and,*

b) *Combine RSP with m-Score of every node to derive a community tree.*

2.2 Calculating m-Score

To calculate the m-Score for every node, we need to investigate each node's importance in a network. Those nodes that link many important nodes are also themselves important. Such a process is very similar to PageRank based algorithms [40]. PageRank is a link analysis algorithm that produces a global "importance" ranking for every web page by analyzing

links among web pages. A fast and efficient page ranking mechanism for web crawling and retrieval remains a challenging issue. Recently, several link based ranking algorithms like PageRank, HITS, OPIC and etc. [27] have been proposed.

We propose a novel method, Correlation Density Rank, based on finding more frequent and influential RSP. The CDR considers the entropy of distance between nodes as punishment and is used to compute ranks of nodes. Hence, there will be a larger traffic amongst shortest path of nodes, if the distance becomes smaller. As proved in [41], if the distance between i and j was less than the distance between i and k, then, i's rank effect on j is more than on k, in other words, the probability that a random surfer reach j from i is more than the probability to reach k. Therefore, the objective is to minimize punishment so that a node with less distance entropy to have a higher rank.

The Shannon entropy is a measurement of system uncertainty [42]. The larger the Shannon entropy is, the more uncertainty the system will be. If the CDR value of a node in complex network is the smallest, then the uncertainty of its distance distribution from other nodes is the greatest. On the contrary, while the CDR value of a node in complex network is very high, then the uncertainty of its distance distribution from other nodes is the smallest.

Moreover, the more popular nodes are, the more linkages other nodes tend to have to them or are linked to by them. The proposed algorithm is analogous to the weighted PageRank algorithm [43], assigning larger rank values to more important (popular) nodes instead of dividing the rank value of a node evenly among its out-link nodes. We assign each out-link node a value proportional to its popularity (its number of in-links and out-links). The popularity from the number of in-links and out-links is recorded as W_{ij}^{in} and W_{ij}^{out}, respectively.

W_{ij}^{in} is the weight of link between node n_i and n_j calculated based on the number of in-links of node n_j and the number of in-links of all reference nodes of node n_i.

$$W_{ij}^{in} = \frac{I_{n_j}}{\sum_{p \in R(n_i)} I_p}$$

Where I_{n_j} and I_p represent the number of frequency in-links of node n_j and node p, respectively. R(n_i) denotes the reference node list of node n_i.

W_{ij}^{out} is the weight of link between node n_i and n_j calculated based on the number of frequency out-links of node n_j and the number of out-links of all reference nodes of node n_i.

$$W_{ij}^{out} = \frac{O_{n_j}}{\sum_{p \in R(n_i)} O_p}$$

Where O_{n_j} and O_p represent the number of out-links of node n_j and node p, respectively. R(n_i) denotes the reference node list of node n_i. These equations has two exceptions, first, if node n_j is a dead-end (which may be easily determined from the frequency matrix), we let $W_{ij}^{out} = \varepsilon$ that ε is a very small number less than 1. Second, $W_{ij}^{out}, W_{ij}^{in} = 1$, that means R($n_i$)=$\{n_j\}$ we add ε to sum of the reference nodes' frequency out/in-link. An algorithm for calculating the m-Score of members in a social network is described as follows.

Algorithm 1. Calculating m-Score for members: Correlation Density Rank (CDR)
Input: social network G
Out: vector of m-Score for all members R

1. Initialize cost distance matrix C

$$C[i, j] = \log_{(1-w_{ij}^{in} w_{ij}^{out})}^{(1-\exp(-\gamma f_{ij}))}$$

(The logarithm of $(1-\exp(-\gamma f_{ij}))$ based on $(1-w_{ij}^{in} w_{ij}^{out})$)

2. Finding the matrix of RSP dissimilarities by employ the algorithm of [43]: {

$W \leftarrow P^{ref} \circ \exp(-\beta C)$

$Z \leftarrow (I - W)^{-1}$

(Note that $(I-W)^{-1} \approx I + W + W^2 + W^3 + \ldots$)

$S \leftarrow \left(Z\left(C \circ W\right)Z\right) \div (Z + \varepsilon)$

$\tilde{C} \leftarrow S - ed_s^T$

$\Delta^{RSP} \leftarrow \lambda\tilde{C} + (1-\lambda)\tilde{C}^T \qquad 0 \le \lambda \le 1$ }

3. M ← Normalize matrix Δ^{RSP} on rows

4. For each node $n_i\ (1 \le i \le k)$ compute the entropy of related row from matrix M:

$$E_i \leftarrow -\frac{1}{Lnk}\sum_{j=1}^{k} M_{ij} Ln(M_{ij})$$

$$d_i \leftarrow 1 - E_i$$

$$R_i \leftarrow d_i \Big/ \sum_{i=1}^{k} d_i$$

5. Return R

ISBN: 978-1-62561-000-3

Where f_{ij} is the number of frequency from node n_i to node n_j. if $f_{ij} = 0$, let $C[i,j] = \infty$ or a very big number. P^{ref} is the transition probability matrix that P_{ij}^{ref} is equal to the rate of f_{ij} divided by sum of frequency between node n_i and all its references nodes. k is the number of members in social network (or nodes on G).

The parameters γ, β and λ are input values determined by user. γ controls the effect of frequency on the cost function which restrict cost ratio with respect to our defined infinite constant. β is the influence of the cost on the walker's selection of a path, and is equal to inverse of temperature at Helmholtz free energy in thermodynamical system [39]. λ is the weight factor by which the leadership status of members will be distinguished by the amount of contact density comes in or goes out.

Also, in step 2, $d_s = diag(S)$ is the vector of diagonal elements of S, and e is the identity matrix. Note that A ∘ B and A ÷ B are elementwise product and division, respectively.

For calculating step 2, we use the easier way of computing the matrix Z [38]. The values of R_i $(1 \le i \le k)$ indicate the final m-Scores of members in the social network.

2.3 Deriving the Community Tree

The m-Score of every member is combined with the normalized RSP matrix (M) on the graph to derive the community tree from an SN. The RSP matrix helps us to find the most likely and closest interaction for each node, and m-Scores determine whether there is the leadership relation between two nodes with closest interaction.

Algorithm 2. Deriving Community Tree: CT_Deriving

Input: Social network G

Output: Community Tree CT

1. CT← [null,…,null]
2. R ← Correlation Density Rank (G);
3. For each member n_i {
   k← $\arg_j \min\ (M_{ij})$
   if R[k] > R[i]
   CT[i] ← k
   }
4. Return CT

Initiating a random walks at node i, we can find ending node j with the most likely correlation density, from the normalized RSP matrix M, by starting. If the m-Score of node j is much greater than that of node i, we regard j as the parent node of node i. After all the operations end, we obtain a community tree CT represented in an array where CT[i] indicates the parent node of node i. A null value of CT[i] indicates that there is no the immediate leader of node i. Hence node i is the core of community. If a node does not have both immediate leader and belongingness, is called private node.

Node i and its descendent compose a community. The m-Score of each member shows the member's importance in the community. The parent of each node is its immediate leader.

3. Exploring evolution of the organizational structure in a dynamic social network

In Section 2, we develop the algorithms to derive a static community tree from a static SN. However, the static community tree does not do a good job in presenting the evolution of the organizational structure in the dynamic SN, because it does not consider intra time-step evolutions.

To present the evolution of organizational structure in a dynamic SN, we aggregate the change of an SN during different time periods, and then derive community trees. We further construct the evolving community tree from the two closest community trees. Hence, the evolving community tree can accurately present evolution of organizational structure over time.

The Best-first search algorithm explores a graph by expanding the most promising node chosen according to a specified rule [44]. Motivated by the idea of the Best-first search, [26] uses tree edit distance [45] (defined as the least cost of edit operation to change a tree to another tree) as a measure of distance (similarity) between the two trees. We propose a new algorithm to derive the evolving community tree with more efficiency and less complexity than previous ways. Because constructing community tree in our idea is based on RSP, we only need to compute linear combination of RSP of two static community trees and use the new RSP to drive evolving CT using algorithm (2) without any iteration. However, [26] generated a collection of candidate evolving community trees and then chose the candidate having the minimum ES score (by means of scoring function, which measures distance errors among evolving CT, previous period CT and current period CT) as a solution of each iteration.

3.1 Tree learning algorithm

We propose a tree learning algorithm to derive an evolving community tree from two static community trees. The constructing process is as follows:

(1) Obtain a collection of members in the evolving community tree $N_{ce} = N_{pre} \bigcup N_{cs}$ where N_{pre} and N_{cs} are collections of members in the previous time period static community tree and current time periods one respectively;

(2) Compute the RSP for all pair members in the evolving community tree, where α is a smoothing factor, $\Delta_{ce}^{RSP} = (1-\alpha) \bullet \Delta_{pre}^{RSP} + \alpha \bullet \Delta_{cs}^{RSP}$. For those members that

appear in the evolving community tree but not in the current community tree, if their m-scores are less than a threshold θ, we regard them as retired members and remove them from the evolving community tree;

(3) According to the definition of algorithm 1, having Δ_{ce}^{RSP} matrix, we can compute R_{ce}, M_{ce} matrices. Then, by applying them into Algorithm 2, we construct the evolving community tree.

Algorithm 3. Learning evolving community tree: ECT_Learning

Input: RSP matrices Δ_{pre}^{RSP} , Δ_{cs}^{RSP}

Output: Evolving community tree CT e

1. $N_{ce} \leftarrow N_{pre} \bigcup N_{cs}$
2. $\Delta_{ce}^{RSP} \leftarrow (1-\alpha) \bullet \Delta_{pre}^{RSP} + \alpha \bullet \Delta_{cs}^{RSP}$
3. $N_{ce} \leftarrow N_{ce} - \{n_i \in N_{ce} - N_{cs} \,|m - score(n_i) < \theta\}$
4. Do step 3,4 of algorithm 1 to find R_{ce}, M_{ce}
5. Employ algorithm2 to find CT_{ce}
6. Return CT_{ce}

3.2 Exploring dynamic social network

After deriving a series of evolving community trees, we exploit these trees to discover the evolution path of the organizational structure and to study the properties of the dynamic SN.

We can consider four types of relationships among communities to generate an evolution graph that represent evolution of the organizational structure. Fig. 2 provides examples to illustrate the relationships among communities [26].

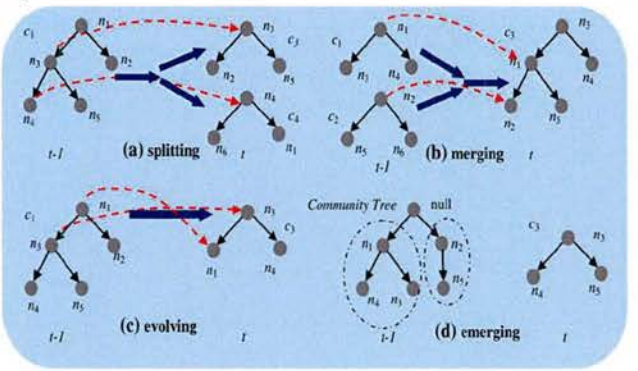


Figure 2. Relationships between communities.

Combing evolution graph and dynamic SN properties, we can obtain insights into the dynamic SN. Definitions 2–4 define key properties of the dynamic SN [26].

Definition 2. (life-line): Let $C = \{c_1,...,c_n\}$ be a collection of communities. For each community $c_i \in C$ with $i < n$, c_{i+1} is the evolving entity of c_i. We say $\{c_1,...,c_n\}$ is a life-line of the community c_1. Community $c_k \in C$ is called the entity of the life-line. A life-line depicts an evolving process of one community in the dynamic SN.

Definition 3. (supporter): Given a life-line $CL = \{c_1, c_2, ..., c_n\}$, we call the members as supporters of CL if they appear in CL not less than δ times with $\delta <= n$.

In Definition 3, δ is a parameter set by the user. If there is a life-line $CL = \{c_1, c_2, c_3, c_4\}$ and $\delta = 3$, that means only members appearing in the life-line not less than 3 times are supporters of CL. Exploring life-lines in the evolution graph and their supporters helps to better understand dynamic social networks. For example, we can discover the backbone of criminal group or detect loyal members in a forum over time.

Definition 4. (Activeness of community): Let node p be the core of community c, we use the m-Score of p to indicate the activeness of community c. Activeness of community is a metric for the extent to which associations occur among members of the community over one period of time. We can use Activeness of community to reveal hot communities, which may reflect on-going hot topics in the forum or new activities in a criminal group.

4. Experiments

In order to implement our approach, first, we consider a small dynamic, synthetic network (Fig. 3) during five time intervals.

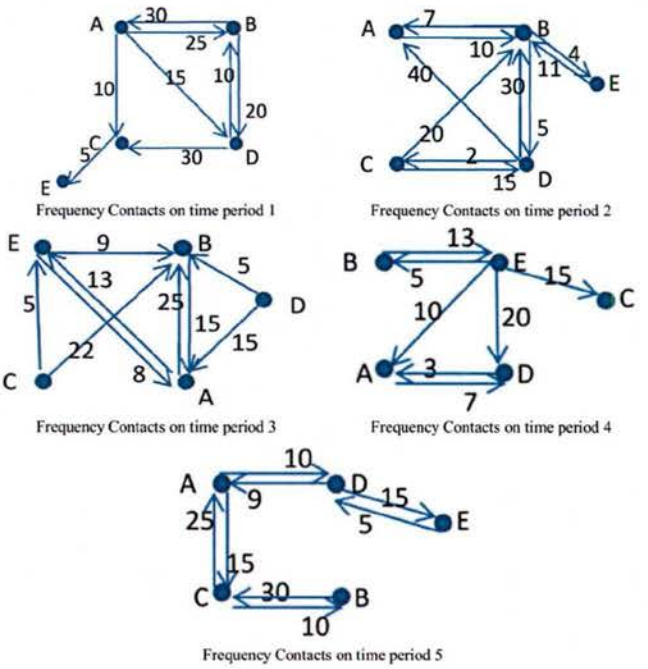


Figure 3. A small dynamic network with frequency contacts between nodes

ISBN: 978-1-62561-000-3

After employing algorithms 1 and 2, we find static community trees for each periods of time separately that shown in Fig. 4 The arrows indicate leadership, and The parameters let $\gamma = 0.1, \varepsilon = 0.001, \lambda = 0.5, \beta = 0.01, \alpha = 0.7$.

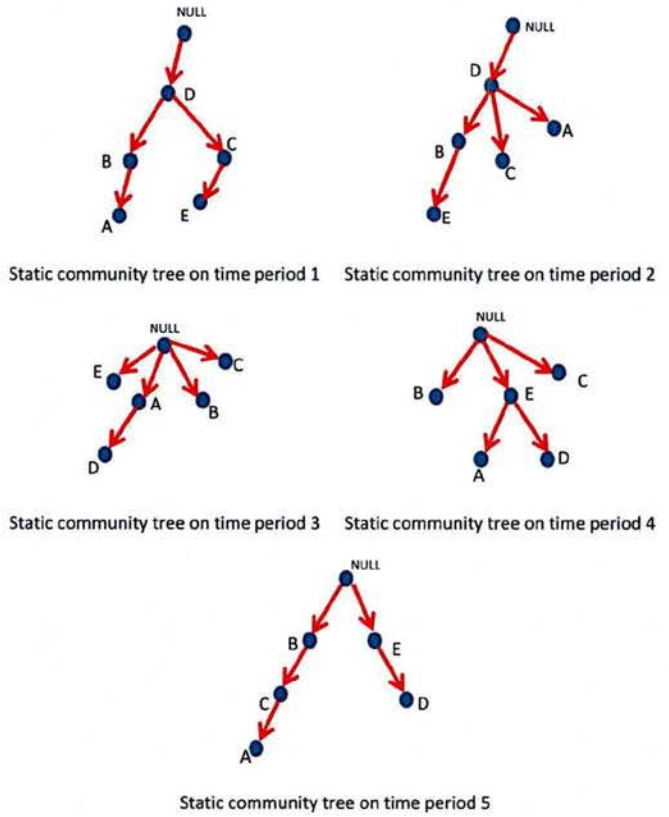


Figure 4. Static community trees for each periods of time separately.

Also, the trend of member's activity in Fig. 5 May be seen.

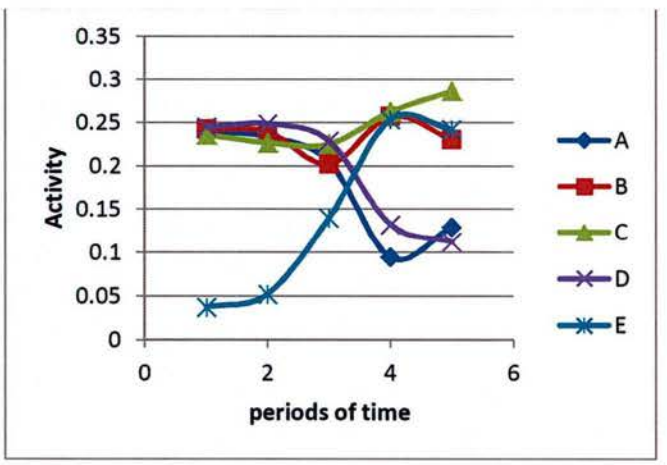


Figure 5. Evolving member's activity for the synthetic network

Now, in this stage, we have information about leadership during each periods of time separately that is not enough for the complete analysis. So, with the purpose of achieve a good understanding of the network's changing trends during intra time-step evolutions, we employ algorithms 3 to drive the evolving community trees shown in Fig. 6.

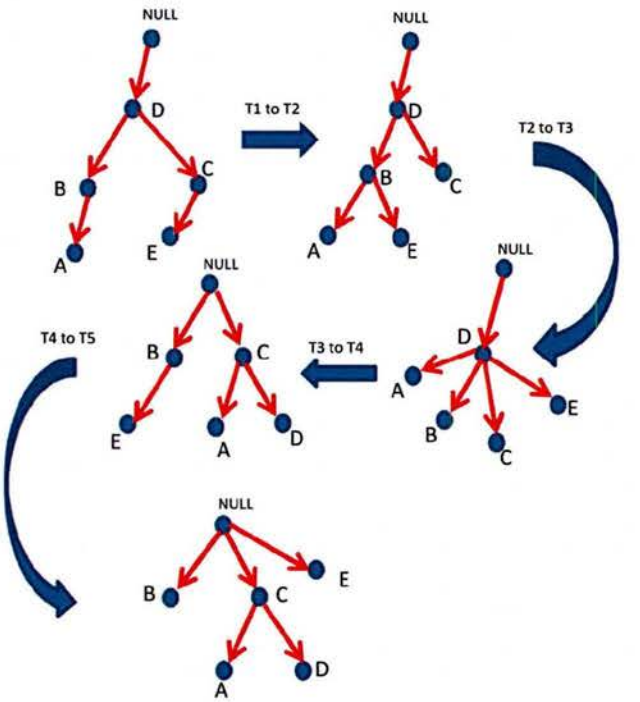


Figure 6. The evolving community trees for the synthetic network

The analysis of outputs clarifies some events that were happened during time:

a) From T1 to T2, the leader of member E changes from C to B.

b) From T2 to T3, leader of members A and E change from B to D.

c) From T3 to T4, 1) splitting on B, C. 2) evolving between C, D. 3) emerging the core D related to members B, E.

d) From T4 to T5, splitting on member E.

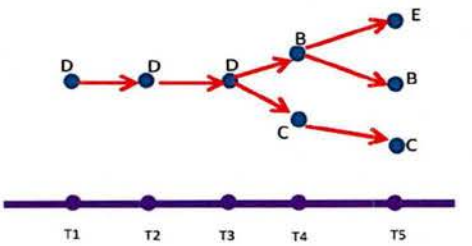


Figure 7. Evolution of communities for the synthetic network.

Fig. 7 Shows the evolution map of communities in which Community that its core is D has most stability (number of

ISBN: 978-1-62561-000-3

supporters divided by the size, named stability. Size is the average number of members in each community in life-lines.) among all communities and its supporters are members B and C.

While the straightforward application of this method is in social networks, this technique is appropriate for all type of complex networks, and the type of network does not influence the results. We have used the real frequency data from a computer network of 288 nodes to evaluate this approach. Data for a period of 100 seconds, divided into 10 equal periods of 10 seconds each, is used to construct evolving community trees and draw evolution map of communities they are.

After employing algorithms 1, 2 and 3, we have reached evolving community trees as shown in Appendix. The arrows indicate hierarchical leadership between nodes, and the parameters let $\gamma = 0.1, \varepsilon = 0.001, \lambda = 0.5, \beta = 0.01, \alpha = 0.7$.

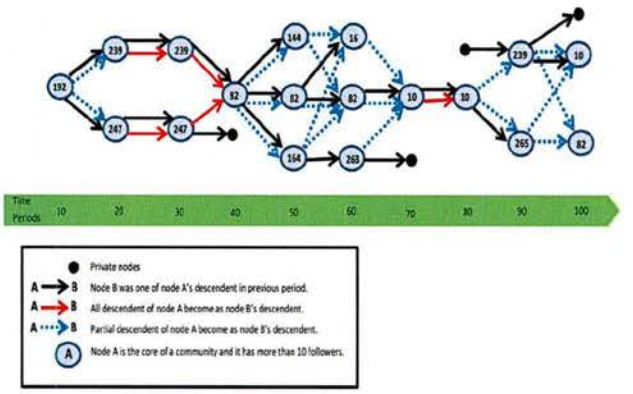


Figure 8. Evolution of communities for the real data set.

Fig. 8 Shows the evolution map of communities, in which the community that its core is node 82 has 185 size, longer life-line (30 second) and the largest stability with amount of 0.7 among all communities. As expected, the algorithm identifies routers and hubs as the cores of communities.

5. Conclusions

Exploring organizational structure in a dynamic social network has a broad range of applications, such as monitoring gang activities, fraud detection, and improving performance of viral marketing. In this paper, we present our research effort in extracting organizational structure from such data to obtain a better understanding of the social network. We formalize a community tree data structure for the purpose of representing the social network organizational structure, and propose a framework to explore the dynamic behavior of the participants of the community. The framework is composed of three main parts: (1) defining the "Correlation Density Rank", to rank the nodes to acquire a community tree from the static social network; and, (2) a tree learning algorithm, which employs the tree edit distance as a scoring function, to generate the evolving community tree; These algorithms were applied to a synthetic and a real world datasets, and produce good results. The experiments show that the framework can well present the organizational structure of a social network.

We obtained experimentally the following insights: (1) those communities with long life-line and great stability likely correspond to a real organization; and (2) the cores in an organizational structure, in general, are either the leaders of the organization or the agents of these leaders. Although it is possible that the organizational structure discovered from a social network is not perfectly in line with the real world organization, the approach described here helps reach new understandings of the organization based on the power of attracting information flow and the interaction closeness.

References

[1] Chekuri, C.S., et al. Experimental study of minimum cut algorithms. in Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms. 1997. Society for Industrial and Applied Mathematics.

[2] Ding, C.H., et al. A min-max cut algorithm for graph partitioning and data clustering. in Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on. 2001. IEEE.

[3] Hagen, L. and A.B. Kahng, New spectral methods for ratio cut partitioning and clustering. Computer-aided design of integrated circuits and systems, ieee transactions on, 1992. 11(9): p. 1074-1085.

[4] Long, B., et al. Community learning by graph approximation. in Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on. 2007. IEEE.

[5] Newman, M.E., Fast algorithm for detecting community structure in networks. Physical review E, 2004. 69(6): p. 066133.

[6] Newman, M.E. and M. Girvan, Finding and evaluating community structure in networks. Physical review E, 2004. 69(2): p. 026113.

[7] Shi, J. and J. Malik, Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2000. 22(8): p. 888-905.

[8] Wu, A.Y., M. Garland, and J. Han. Mining scale-free networks using geodesic clustering. in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. 2004. ACM.

[9] Xu, X., et al. SCAN: a structural clustering algorithm for networks. in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. 2007. ACM.

[10] Zhou, D., et al. Probabilistic models for discovering e-communities. in Proceedings of the 15th international conference on World Wide Web. 2006. ACM.

[11] Goyal, A., F. Bonchi, and L.V. Lakshmanan. Discovering leaders from community actions. in Proceedings of the 17th ACM conference on Information and knowledge management. 2008. ACM.

[12] Xu, J.J. and H. Chen, CrimeNet explorer: a framework for criminal network knowledge discovery. ACM Transactions on Information Systems (TOIS), 2005. 23(2): p. 201-226.

[13] Carley, K.M., et al., Toward an interoperable dynamic network analysis toolkit. Decision Support Systems, 2007. 43(4): p. 1324-1347.

[14] [14]. Ma, H., et al. Mining social networks using heat diffusion processes for marketing candidates selection. in Proceedings of the 17th ACM conference on Information and knowledge management. 2008. ACM.

[15] Xu, J.J. and H. Chen, Fighting organized crimes: using shortest-path algorithms to identify associations in criminal networks. Decision Support Systems, 2004. 38(3): p. 473-487.

[16] Kumar, R., J. Novak, and A. Tomkins, Structure and evolution of online social networks, in Link mining: models, algorithms, and applications. 2010, Springer. p. 337-357.

[17] Tang, L., et al. Community evolution in dynamic multi-mode networks. in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 2008. ACM.

[18] Tantipathananandh, C., T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. 2007. ACM.

[19] [19]. Zhou, D., et al. Discovering temporal communities from social network documents. in Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on. 2007. IEEE.

[20] Bertino, E., E. Ferrari, and V. Atluri, The specification and enforcement of authorization constraints in workflow management systems. ACM Transactions on Information and System Security (TISSEC), 1999. 2(1): p. 65-104.

[21] Song, M. and W.M. van der Aalst, Towards comprehensive support for organizational mining. Decision Support Systems, 2008. 46(1): p. 300-317.

[22] Zur Muehlen, M., Organizational management in workflow applications–issues and perspectives. Information Technology and Management, 2004. 5(3-4): p. 271-291.

[23] Clauset, A., C. Moore, and M.E. Newman, Hierarchical structure and the prediction of missing links in networks. Nature, 2008. 453(7191): p. 98-101.

[24] Grobelnik, M., D. Mladenic, and B. Fortuna, Semantic technology for capturing communication inside an organization. Internet Computing, IEEE, 2009. 13(4): p. 59-67.

[25] Li, H., et al. Scalable community discovery on textual data with relations. in Proceedings of the 17th ACM conference on Information and knowledge management. 2008. ACM.

[26] Qiu, J. and Z. Lin, A framework for exploring organizational structure in dynamic social networks. Decision Support Systems, 2011. 51(4): p. 760-771.

[27] Singh, A.K. and P. Ravi Kumar, A comparative study of page ranking algorithms for information retrieval. International journal of electrical and computer engineering, 2009. 4(7): p. 469-480.

[28] Rajaraman, A. and J.D. Ullman, Mining of massive datasets. 2012: Cambridge University Press.

[29] Cho, J., S. Roy, and R.E. Adams. Page quality: In search of an unbiased web ranking. in Proceedings of the 2005 ACM SIGMOD international conference on Management of data. 2005. ACM.

[30] Jin, Hong, Shuliang Wang, and Chenyang Li. "Community detection in complex networks by density-based clustering." Physica A: Statistical Mechanics and its Applications 392.19 (2013): 4606-4618.

[31] Qi, X., et al., Optimal local community detection in social networks based on density drop of subgraphs. Pattern Recognition Letters, 2014. 36: p. 46-53.

[32] Gong, M., et al., Novel heuristic density-based method for community detection in networks. Physica A: Statistical Mechanics and its Applications, 2014. 403: p. 71-84.

[33] Song, M. and W.M. van der Aalst, Towards comprehensive support for organizational mining. Decision Support Systems, 2008. 46(1): p. 300-317.

[34] Gruhl, D., et al. Information diffusion through blogspace. in Proceedings of the 13th international conference on World Wide Web. 2004. ACM.

[35] Craswell, N. and M. Szummer. Random walks on the click graph. in Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. 2007. ACM.

[36] Lovász, L., Random walks on graphs: A survey. Combinatorics, Paul erdos is eighty, 1993. 2(1): p. 1-46.

[37] Yen, L., et al. A family of dissimilarity measures between nodes generalizing both the shortest-path and the commute-time distances. in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 2008. ACM.

[38] Saerens, M., et al., Randomized shortest-path problems: Two related models. Neural Computation, 2009. 21(8): p. 2363-2404.

[39] Kivimäki, I., M. Shimbo, and M. Saerens, Developments in the theory of randomized shortest paths with a comparison of graph node distances. Physica A: Statistical Mechanics and its Applications, 2014. 393: p. 600-616.

[40] Brin, S. and L. Page, The anatomy of a large-scale hypertextual Web search engine. Computer networks and ISDN systems, 1998. 30(1): p. 107-117.

[41] Zareh Bidoki, A.M. and N. Yazdani, DistanceRank: An intelligent ranking algorithm for web pages. Information Processing & Management, 2008. 44(2): p. 877-892.

[42] Anand, K. and G. Bianconi, Entropy measures for networks: Toward an information theory of complex topologies. Physical Review E, 2009. 80(4): p. 045102.

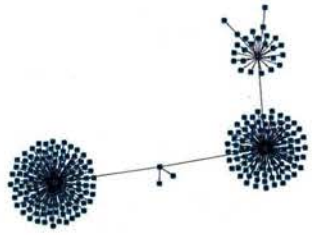
[43] Xing, W. and A. Ghorbani. Weighted pagerank algorithm. in Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on. 2004. IEEE.

[44] Pearl, J., Heuristics: intelligent search strategies for computer problem solving. 1984.

[45] Bille, P., A survey on tree edit distance and related problems. Theoretical computer science, 2005. 337(1): p. 217-239.

Appendix

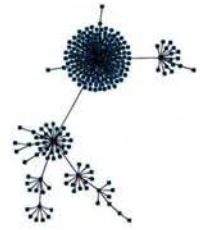
Time Period 1



Evolving between Time Periods 1 and 2

Evolving between Time Periods 2 and 3

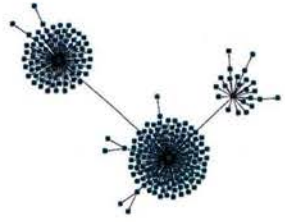
Evolving between Time Periods 3 and 4



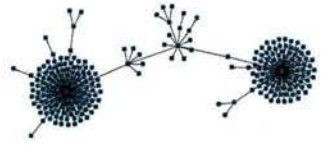
Evolving between Time Periods 4 and 5

Evolving between Time Periods 5 and 6

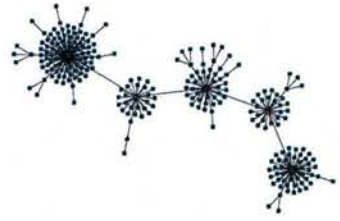
Evolving between Time Periods 6 and 7



Evolving between Time Periods 7 and 8



Evolving between Time Periods 8 and 9



Evolving between Time Periods 9 and 10

The evolving community trees

ISBN: 978-1-62561-000-3

Network Performance Rank: An Approach for Comparison of Complex Networks

Zeynab Bahrami Bidoni [1], Roy George [2]
Department of Computer and Information Systems
Clark Atlanta University, Atlanta, GA
z.bahrami62@gmail.com [1]
rgeorge@cau.edu [2]

Abstract

Researchers have typically concentrated on analyzing what happens internally in a complex network and using this to distinguish between nodes. However, there has been less effort towards comparing between different networks. In this paper, we proposed a novel approach to rank alternative complex networks based on their performances. We consider this as a ranking problem in decision analysis based on occurring positive/negative frequent events as criteria, and using the TOPSIS method to rank alternatives. In order to assign a score to the networks for each criterion, a statistical method that estimates the expected value of positive/negative frequent events on a random node is presented. The proposed technique is efficient in terms of algorithm complexity and is capable of discriminating events occurring between important nodes over those between less significant nodes. The experiments, conducted on several synthetic networks, demonstrate the feasibility and applicability of the ranking methodology.

Keywords: Complex Network; Network Performance Rank (NPR); Correlation Density Rank (CDR); Multi-Criteria Decision Making (MCDM); TOPSIS method; Renyi entropy; Gaussian influence function.

1. Introduction

In current years, researchers have mostly focused on the internals of complex networks developing techniques such as detecting communities [1-12], ranking nodes [13-18], finding outliers [19-21], and etc. There has been less attention given towards the performance comparisons between different networks. This problem manifests itself in different domains including Computer, Telecommunication, Electrical Circuit, Supply Chain, Social networks etc., where, there is a need to evaluate different network architectures, equipment, protocols etc. with the constraint, that it is not possible to replicate the exact same scenarios in each case. In this study, we assume this objective as a Ranking problem in Multi-Criteria Decision Making (MCDM) [22-26] field based on occurring positive/negative frequent events as the criteria.

Since any event occurs between two nodes of a network, and the nodes could not considered as independent variables, statistical analysis to compute the probability of failed/successful occurrence between random nodes throughout the network would be very difficult. This paper proposes a novel approach to approximate variance of all type of event per networks, which is used to estimate the expected values of the events between two random nodes.

The contributions of this paper are as follows: (1) Defining the networks performance comparison problem as a Multi-Criteria Decision Making (MCDM) ranking problem, (2) Developing an approach to compute the diversity of density (DOD) of events in networks to evaluate the variance where the events happening between important nodes are positively discriminated over events between less significant nodes. (3) Approximating the probability distribution and the expected value of occurrences on a random node for scoring each network per criteria.

The rest of the paper is organized as follows. Section 2 presents the general framework and all approaches needed for ranking alternative networks. Section 3 provides the experimental results. Section 4 offers concluding remarks.

2. Proposed approach to compare between networks

2.1 General framework

In order to compare between networks performance, we consider this issue as a ranking problem in MCDM. A MCDM problem can be concisely expressed in matrix format as

	C_1	C_2	...	C_n
A_1	x_{11}	x_{12}	...	x_{1n}
A_2	x_{21}	x_{22}	...	x_{2n}
A_m	x_{m1}	x_{m2}	...	x_{mn}

$$W = [w_1, w_2, \ldots, w_n]$$

Figure 1. A decision matrix in MCDM problem model

Where $A_1, A_2, ..., A_m$ are possible alternative networks among which have to rank, $c_1, c_2, ..., c_n$ are criteria with which alternative performance are measured, x_{ij} is the score of alternative A_i with respect to criterion c_j, w_j is the weight of criterion c_j.

Various attributes can be selected as criteria but, here, we focus on positive/negative frequent events which may occur between nodes during a given big enough period of time, and effect on network's performance. For instance, the re-transaction or any failed operation between nodes is the negative event in net which decrease network's efficiency. On the contrary, the more probability density of successful and positive occurrence is the more efficiency will be in the network.

This research is funded in part by the Army Research Laboratory under Grant No: W911NF-12-2-0067 and Army Research Office under Grant Number W911NF-11-1-0168. Any opinions, findings, conclusions or recommendations expressed here are those of the author(s) and do not necessarily reflect the views of the sponsor.

So, for establishing the decision matrix follow steps needed:

a) Select the collection of criteria.

b) Scoring networks on criteria.

After that, using TOPSIS method which explained in section 2-3 to rank the alternative networks.

2.2 Scoring networks on criteria

With the purpose of scoring networks on criteria, we proposed a new density based approach which compute the global DOD of the given positive/negative frequent event (criteria) for each networks. Gaussian distribution is employed based on the average number of events per unit as the mean parameter and the approximated DOD is used as the variance parameter to estimate the expected value of the event frequency between two random nodes per network during given big enough time periods.

In order to compute the global DOD on given criterion, we used the modified ''Correlation Density Rank'' Method [27] which finds probability density distribution of the related frequent event on all nodes, and then we utilize the Renyi entropy [28] to realize the global unpredictability or diversity of these densities on whole network.

2.2.1.Correlation Density Rank

We use the Correlation Density Rank (CDR), [27] which finds more frequent and influential Randomized shortest Path (RSP). The CDR considers the distance between nodes as punishment and is used to compute probability density of nodes. Hence, there will be a larger traffic amongst shortest path of nodes, if the distance becomes smaller. Therefore, the objective is to minimize punishment so that a node with high value of density probability to have a higher rank.

Moreover, the more popular nodes are the more linkages other nodes tend to have to them or are linked to by them. The proposed algorithm is analogous to the weighted PageRank algorithm [29, 30], assigning larger rank values to more important (popular) nodes instead of dividing the rank value of a node evenly among its out-link nodes. We assign each out-link node a value proportional to its popularity (its number of in-links and out-links). The popularity from the number of in-links and out-links is recorded as W_{ij}^{in} and W_{ij}^{out}, respectively.

W_{ij}^{in} is the weight of link between node n_i and n_j calculated based on the number of in-links of node n_j and the number of in-links of all reference nodes of node n_i.

$$W_{ij}^{in} = \frac{I_{n_j}}{\sum_{p \in R(n_i)} I_p} \tag{1}$$

Where I_{n_j} and I_p represent the number of frequency in-links of node n_j and node p, respectively. R(n_i) denotes the reference node list of node n_i.

W_{ij}^{out} is the weight of link between node n_i and n_j calculated based on the number of frequency out-links of node n_j and the number of out-links of all reference nodes of node n_i.

$$W_{ij}^{out} = \frac{O_{n_j}}{\sum_{p \in R(n_i)} O_p} \tag{2}$$

Where O_{n_j} and O_p represent the number of out-links of node n_j and node p, respectively. R(n_i) denotes the reference node list of node n_i. These equations has two exceptions, first, if node n_j is a dead-end (which may be easily determined from the frequency matrix), we let $W_{ij}^{out} = \varepsilon$ that ε is a very small number less than 1. Second, $W_{ij}^{out}, W_{ij}^{in} = 1$, that means R($n_i$)=$\{n_j\}$ we add ε to sum of the reference nodes' frequency out/in-link.

An algorithm for calculating the probability density of related frequent event for all members in a complex network is described as follows.

Algorithm 1. Correlation Density Rank (CDR):

Input: social network G

Out: vector of probability density distribution CDR

a) Initialize cost distance matrix C

$$C[i,j] = \log_{\left(1-w_{ij}^{in} w_{ij}^{out}\right)}^{(1-\exp(-\gamma f_{ij}))} \tag{3}$$

(The logarithm of $(1-\exp(-\gamma f_{ij}))$ based on $(1-w_{ij}^{in} w_{ij}^{out})$)

b) Finding the matrix of RSP dissimilarities by employ the algorithm of [29]:

{

$$W \leftarrow P^{ref} \circ \exp(-\beta C) \tag{4}$$

$$Z \leftarrow (I - W)^{-1} \tag{5}$$

(Note that $(I-W)^{-1} \approx I + W + W^2 + W^3 + \ldots$)

$$S \leftarrow \left(Z \left(C \circ W\right) Z\right) \div (Z + \varepsilon) \tag{6}$$

$$C \leftarrow S - e d_s^T \tag{7}$$

$$\Delta^{RSP} \leftarrow 0.5(\bar{C}+\bar{C}^{T}) \quad (8)$$

}

c) $M \leftarrow$ Normalize matrix Δ^{RSP} on columns

d) For each node nj ($1 \le j \le k$) compute inverse of the entropy [31] of related column from matrix M (σj is the jth kernel scale parameter which describes the influence of a node nj within its Neighborhood. we optimize σ for each node to make the density values the most different):

$$ej \leftarrow -\frac{1}{Lnk}\sum_{i=1}^{k} M_{ij} Ln(M_{ij}) \quad (9)$$

$$\sigma j \leftarrow \frac{1}{e_j} \quad (10)$$

e) Calculate the density function which results from a Gauss Influence function [32] (it sorts all the nodes in descending order according to their CDR values)

$$cdr_i \leftarrow \sum_{j=1}^{k} \exp\left(-\frac{\left(\Delta_{ij}^{RSP}\right)^2}{2\sigma_j^2}\right) \quad (11)$$

f) Normalize Correlation Density Rank vector (we can sort all the nodes in descending order according to their CDR values):

$$CDR_i \leftarrow {cdr_i}\Big/{\sum_{i=1}^{k} cdr_i} \quad (12)$$

g) Return CDR.

Where f_{ij} is the number of frequency from node n_i to node n_j. if $f_{ij} = 0$, let $C[i,j] = \infty$ or a very big number. P^{ref} is the transition probability matrix that P_{ij}^{ref} is equal to the rate of f_{ij} divided by sum of frequency between node n_i and all its references nodes. k is the number of members in social network (or nodes on G).

The parameters γ and β are input values determined by user. γ controls the effect of frequency on the cost function which restrict cost ratio with respect to our defined infinite constant. β is the influence of the cost on the walker's selection of a path, and is equal to inverse of temperature at Helmholtz free energy in thermodynamical system [33].

Also, in step 2, $d_s = diag(S)$ is the vector of diagonal elements of *S*, and *e* is the identity matrix. Note that A ∘ B and A ÷ B are elementwise product and division, respectively.

For calculating step 2, we use the easier way of computing the matrix Z [34]. The values of CDR_i ($1 \le i \le k$) indicate the final normalized density rank of members in the complex network which are considered as probability density distribution on nodes.

2.2.2.Measure of the global unpredictability/ DOD for each criterion per network

The Shannon entropy is a measurement of system uncertainty, unpredictability, diversity and randomness [31] and has been used in statistics and information theory to develop measures of the information content [35]. The larger the Shannon entropy is, the more uncertainty/unpredictability/randomness and less diversity of the system will be. Also, Shannon entropy is the classical measure of information content and is defined for an n-dimensional probability density (PD) distribution P(x) as:

$$H(P) = \int_{-\infty}^{\infty} P(x)\log P(x)dx \quad (13)$$

Since several time frequency representations can achieve negative values the use of the more classical Shannon information as a measure of complexity is prohibited (due to the presence of the logarithm within the integral in below) and some authors [28, 36-38] have proposed the use of a relaxed measure of entropy known as the Renyi entropy of order α :

$$H_{\alpha}^{R}(P) = \frac{1}{1-\alpha}\log\frac{\int P^{\alpha}(x)dx}{\int P(x)dx} \quad (14)$$

Following Baraniuk, the passage from the Shannon entropy H to the class of Renyi entropies H_{α}^{R} involves only the relaxation of the mean value property from an arithmetic to an exponential mean and thus in practice H_{α}^{R} behaves much like H. The Shannon entropy can be recovered as $\lim_{\alpha\to 1} H_{\alpha}^{R}(P) = H(P)$.
(15)

So, in order to measure of the global DOD/unpredictibility for each network, we can employ the CDR vector as the probability density distribution on nodes in Renyi entropy formulate. Thus, for scoring each network on each criteria, we compute the follow measure:

$$H_k^{c_l} = \frac{1}{1-\alpha}\log_2\left(\frac{\sum_{i=1}^{N_k} CDR_i^{\alpha}}{\sum_{i=1}^{N_k} CDR_i}\right) \quad (16)$$

Where $H_k^{c_l}$ the unpredictability of network number k on the event related to criterion c_l and N_k is the number of nodes in network number *K*. Also, *CDR* vector is related to given network and event, and α is the order of Renyi entropy order that we can consider 3.

If the density value of each node in complex network is the same, then the uncertainty of the original density distribution is the greatest. On the contrary, while the density value of each node in complex network is very asymmetrical, then the uncertainty of the original density distribution is the smallest.

Thus, the inverse of uncertainty/unpredictability would be a good measure for the DOD through network.

2.2.3. Estimate the expected value rate of each event per each networks

As mentioned above, the Renyi entropy can reflect the difference of nodes' density value. The more different the density values are, the smaller the Renyi entropy is. So, we can consider inverse of Reyni entropy's result time the mean as a measure of the frequent event variance through nodes.

The normal (or Gaussian) distribution is a very commonly occurring continuous probability distribution—a function that tells the probability that an observation in some context will fall between any two real numbers. Normal distributions are extremely important in statistics and are often used in the natural and social sciences for real-valued random variables whose distributions are not known [39]. Furthermore, considering a Guassian distribution with below mean and variance parameters, can help us to better understanding about probability distribution of given frequent event through kth network.

$$\mu_k^{c_l} = \frac{\sum_{i=1}^{N_k}\sum_{j=1}^{N_k} f_{ij}}{N_k} \quad , \quad \delta_k^{c_l} = \frac{\mu_k^{c_l}}{H_k^{c_l}} \tag{17}$$

$$F_k\left(x, \mu_k^{c_l}, \delta_k^{c_l}\right) = \frac{1}{\delta_k^{c_l}\sqrt{2\pi}} e^{\frac{\left(x-\mu_k^{c_l}\right)^2}{2(\delta_k^{c_l})^2}} \qquad x \geq 0 \tag{18}$$

For instance, probability of not occur the given event on a random node in kth network would be result of below equation:

$$p(x=0) = \frac{e^{-\left(\mu_k^{c_l}\right)^2 / 2(\delta_k^{c_l})^2}}{\delta_k^{c_l}\sqrt{2\pi}} \tag{19}$$

Moreover, the expected value of event on two random nodes in networks' Gaussian distribution is a good measure for scoring networks on related criteria. Thus, we have

$$S_k^{c_l} = \int_0^{+\infty} x F_k\left(x, \mu_k^{c_l}, \delta_k^{c_l}\right) dx = \int_0^{+\infty} \frac{x}{\delta_k^{c_l}\sqrt{2\pi}} e^{\frac{\left(x-\mu_k^{c_l}\right)^2}{2(\delta_k^{c_l})^2}} dx$$

□□□□□

Where $S_k^{c_l}$ is the score of kth network on criterion c_l. After using this approach for all networks and scoring them on all criteria, Decision Matrix is constructed to apply TOPSIS method for ranking networks' performance. Note that before start TOPSIS's steps normalize Decision matrix on rows to have the ratio of expected value for each networks. Because, networks may have different amounts of global expected values of all event types on a random node. So, for fair compare between them, the ratio of expected value is a better measure to evaluate.

2.3 TOPSIS algorithm

Many ranking methods have been proposed to solve the multiple criteria decision making (MCDM) problems, etc. One of the well-known ranking methods for MCDM, named the technique for order preference by similarity to ideal solution (TOPSIS) [40-45], is firstly proposed by Hwang and Yoon [46]. The logic of the TOPSIS approach is to define the ideal and anti-ideal solutions [43], which are based on the concept of relative closeness in compliance with the shorter (longer) the distance of alternative i to ideal (anti-ideal), the higher the priority can be ranked [47]. The procedure of TOPSIS can be expressed in a series of steps:

(1) Calculate the normalized decision matrix on column. The normalized value n_{ij} is calculated as

$$n_{ij} = x_{ij} \Big/ \sqrt{\sum_{i=1}^{m} x_{ij}^2} \qquad j = 1,\ldots,n, \quad i = 1,\ldots,m. \tag{21}$$

(2) Calculate the weighted normalized decision matrix. In this paper, the weights of objective criteria, using the entropy weighting method [47], will be applied to the generalized algorithm. The weighted normalized value v_{ij} is calculated as

$$v_{ij} = w_j n_{ij}, \qquad j = 1,\ldots,n, \quad i = 1,\ldots,m, \tag{22}$$

Where wj is the weight of the ith attribute or criterion, and $\sum_{i=1}^{n} wj = 1$.

(3) Determine the positive ideal and negative ideal solution.

$$A^+ = \left\{v_1^+,\ldots,v_n^+\right\} = \left\{\left(\max_i v_{ij} \,\middle|\, j \in I\right), \left(\min_i v_{ij} \,\middle|\, j \in J\right)\right\},$$

$$A^- = \left\{v_1^-,\ldots,v_n^-\right\} = \left\{\left(\min_i v_{ij} \,\middle|\, j \in I\right), \left(\max_i v_{ij} \,\middle|\, j \in J\right)\right\},$$

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

Where I is associated with benefit criteria, and J is associated with cost criteria.

(4) Calculate the separation measures, using the m-dimensional Euclidean distance. The separation of each alternative from the ideal solution is given as

$$d_i^+ = \left\{\sum_{j=1}^{n}\left(v_{ij} - v_j^+\right)^2\right\}^{\frac{1}{2}}, \qquad i = 1,2,\ldots,m. \tag{24}$$

Similarly, the separation from the negative ideal solution is given as

$$d_i^- = \left\{ \sum_{j=1}^{n} \left(v_{ij} - v_j^- \right)^2 \right\}^{\frac{1}{2}}, \quad i = 1,2,...,m. \tag{25}$$

(5) Calculate the relative closeness to the ideal solution. The relative closeness of the alternative A_j with respect to A^+ is defined as

$$R_i = d_i^- / \left(d_i^+ + d_i^- \right), \quad i = 1,...,m. \tag{26}$$

Since $d_j^- \geq 0$ and $d_j^+ \geq 0$, then, clearly, $R_j \in [0,1]$.

(6) Rank the preference order. For ranking networks using this index, we can rank networks' the relative closeness value in decreasing order.

The basic principle of the TOPSIS method is that the chosen alternative should have the ''shortest distance'' from the positive ideal solution and the ''farthest distance'' from the negative ideal solution. The TOPSIS method introduces two ''reference'' points, but it does not consider the relative importance of the distances from these points.

3. Experiments

In order to implement our approach, we designed four different synthetic architectures of computer network with recording the successful and failed type of frequencies as positive and negative events respectively, during sample time period which are shown on Figure 2. Networks' data can show obviously our method behavior on different situation. For instance, Network A and B have same number of successful and failed events, But the DOD of failed events in network A is higher than network B so that it seems node 8 in network A has a critical problem and probability of happening failed events between node 8 and any other nodes is high.

After employing correlation density rank and Renyi entropy we have found global unpredictability and variance of events per networks, which mentioned in Table 1. within the other general information.

Table 1. Network properties and their unpredictability, mean and variance results by proposed method.

Network Name	Number of nodes	Successful events				Fail events			
		Number of frequency	Unpredictability	Mean	Variance	Number of frequency	Unpredictability	Mean	Variance
Network A	10	100	2.6610	10	3.757	25	0.01	2.5	250
Network B	12	100	3.0501	8.333	2.732	25	1.7552	2.083	1.186
Network C	8	60	2.0871	7.5	3.5935	30	2.1121	3.75	1.7754
Network D	9	60	1.4401	6.6667	4.62933	10	0.7935	1.1111	1.4002

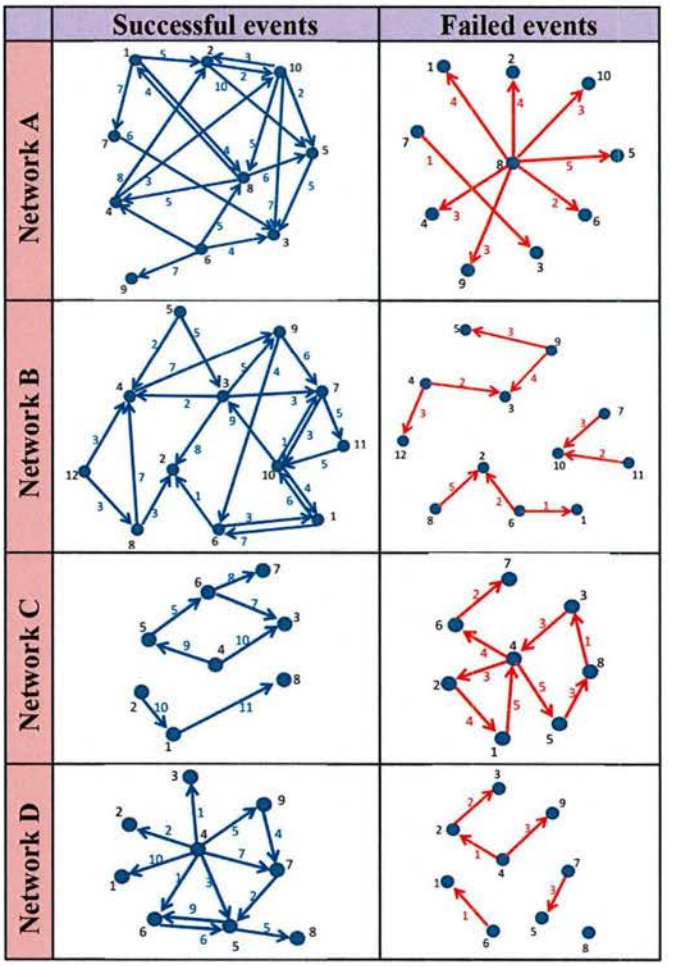


Figure 2. Four synthetic architectures of computer network with their successful and failed frequencies during sample time period.

By having the estimated mean and variance parameters, we can consider networks' probability distributions on both successful and failed events (Figure 3 and Figure 4). For example in Figure 4, the junction of probability distribution curves and Y axis indicate the probability of no occurring failed event on a random node on related network that in this case network D, B, C and A respectively have descending order of the probability of no occurring failed event on a random node.

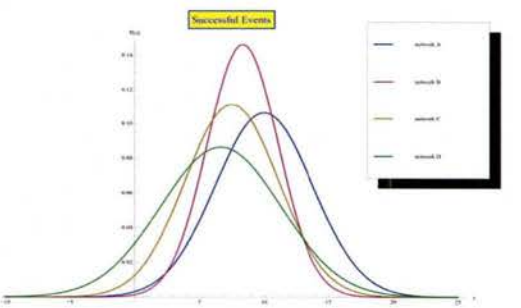


Figure 3. Estimated probability distributions for networks about successful type of events.

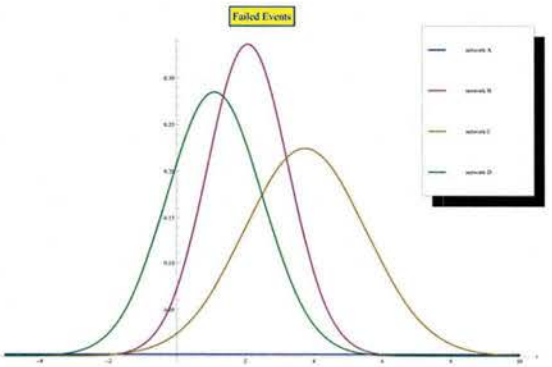


Figure 4. Estimated probability distributions for networks about failed type of events.

In next stage, decision matrix constructed as shown in Table 2 and then using Entropy weighting Method, weights of criteria were computed (Table 3).

Table 2. Decision matrix.

Network Name	Criterion 1 Expected value of successful event	Criterion 2 Expected value of failed event
Network A	10.0045	100.991
Network B	8.3387	2.10188
Network C	7.52409	3.7611
Network D	6.822	1.28134

Table 3. Results of weighting criteria by Entropy weighting method.

Entropy weighting method	Criterion 1 successful event	Criterion 2 failed event
weight	0.6425	0.3575

Finally, the Topsis Method helped us to rank networks based on these two criteria (Table 4).

Table 4. Results of ranking by TOPSIS

Network Name	Network A	Network B	Network C	Network D
Rank value by Topsis method	0	0.942537	0.766967	1

As expected, rank values by TOPSIS method displayed that network D is the best one and networks B, C and A, respectively, have smaller ranks on descending order.

4. Conclusions

Ranking complex networks has a broad range of applications, such as Computer/Corporate/Campus Area Network (CAN), Telecommunication Network, Electrical Circuit Network, Social Network, Supply Chains, Financial networks and etc. In this paper, we present our research effort in comparing between complex networks from their positive/negative frequency data to obtain a ranking of them. The proposed method is composed of three main parts: (1) an approach for estimating the DOD of event frequencies through network; (2) a static framework to explore the expected value of each type of events frequency on a random node per network which is considered as the score of network on related criterion; and, (3) construct the decision matrix and employ the well-known TOPSIS method to rank alternative networks. These algorithms were applied to several synthetic datasets, and produce good results. The experiments show that the framework can well present the rank order of networks.

References

[1] Long, B., Wu, X., Zhang, Z., & Yu, P.S. "Community learning by graph approximation. in Data Mining", *ICDM 2007. Seventh IEEE International Conference on. 2007. IEEE.*

[2] Newman, M.E. "Fast algorithm for detecting community structure in networks". *Physical review E*, 2004. 69(6): p. 066133.

[3] Newman, M.E. and M. Girvan, "Finding and evaluating community structure in networks". *Physical review E,* 2004. 69(2): p. 026113.

[4] Goyal, A., F. Bonchi, and L.V. Lakshmanan. "Discovering leaders from community actions". *Proceedings of the 17th ACM conference on Information and knowledge management*. 2008. ACM.

[5] Kumar, R., J. Novak, and A. Tomkins, "Structure and evolution of online social networks, in Link mining: models, algorithms, and applications". *Springer* 2010, p. 337-357.

[6] Tang, L., et al. "Community evolution in dynamic multi-mode networks". *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008. ACM.

[7] Tantipathananandh, C., T. Berger-Wolf, and D. Kempe. "A framework for community identification in dynamic social networks". *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2007. ACM.

[8] Zhou, D., Council, I., Zha, H., & Giles, C.L. "Discovering temporal communities from social network documents". Data Mining, ICDM 2007. Seventh IEEE International Conference on. 2007. IEEE.

[9] Li, H., et al. "Scalable community discovery on textual data with relations". *Proceedings of the 17th ACM conference on Information and knowledge management*. 2008. ACM.

[10] Jin, Hong, Shuliang Wang, and Chenyang Li. "Community detection in complex networks by density-based clustering." *Physica A: Statistical Mechanics and its Applications* 392.19 (2013): 4606-4618.

[11] Qi, X., et al., "Optimal local community detection in social networks based on density drop of subgraphs". *Pattern Recognition Letters*, 2014. 36: p. 46-53.

[12] Gong, M., et al., "Novel heuristic density-based method for community detection in networks". *Physica A: Statistical Mechanics and its Applications*, 2014. 403: p. 71-84.

[13] Zareh Bidoki, A.M. and N. Yazdani, "DistanceRank: An intelligent ranking algorithm for web pages". *Information Processing & Management,* 2008. 44(2): p. 877-892.

[14] Koschützki, D., Schwöbbermeyer, H., & Schreiber, F. "Ranking of network elements based on functional substructures". *Journal of theoretical biology*,2007, 248(3), 471-479.

[15] O'Madadhain, J., Hutchins, J., & Smyth, P., "Prediction and ranking algorithms for event-based network data". *ACM SIGKDD Explorations Newsletter*, 2005, 7(2), 23-30.

[16] Sun, Y., Yu, Y., & Han, J.," Ranking-based clustering of heterogeneous information networks with star network schema". *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* , 2009, pp. 797-806. ACM.

[17] Walker, D., Xie, H., Yan, K. K., & Maslov, S. "Ranking scientific publications using a model of network traffic". *Journal of Statistical Mechanics: Theory and Experiment*, 2007(06), P06010.

[18] Brown, L. D. "Ranking journals using social science research network downloads". *Review of Quantitative Finance and Accounting*, 2003, 20(3), 291-307.

[19] Gopalan, Prem K., and Bryan Thomas Elverson. "Detecting Outliers in Network Traffic Time Series." *U.S. Patent Application* 12/266,105, 2014.

[20] Zhang, J., & Zulkernine, M. "Anomaly based network intrusion detection with unsupervised outlier detection". *Communications, 2006. ICC'06. IEEE International Conference on* (Vol. 5, pp. 2388-2393). IEEE.

[21] Kotidis, Y., Vassalos, V., Deligiannakis, A., Stoumpos, V., & Delis, A. "Robust management of outliers in sensor network aggregate queries". *Proceedings of the 6th ACM international workshop on Data engineering for wireless and mobile access* , 2007, pp. 17-24. ACM.

[22] Massam, B. H. "Multi-criteria decision making (MCDM) techniques in planning. Progress in planning", 1988, 30, 1-84.

[23] Carlsson, C., & Fullér, R. "Fuzzy multiple criteria decision making: Recent developments". *Fuzzy sets and systems*, 1996, 78(2), 139-153.

[24] Dyer, J. S., Fishburn, P. C., Steuer, R. E., Wallenius, J., & Zionts, S. "Multiple criteria decision making, multiattribute utility theory: the next ten years". *Management science*, 1992, 38(5), 645-654.

[25] Kahraman, C. "Fuzzy multi-criteria decision making: theory and applications with recent developments" *Springer*, 2008, Vol. 16.

[26] Triantaphyllou, E. "Multi-criteria decision making methods", *Springer*, 2000, pp. 5-21.

[27] Bahrami Bidoni, Z., George, R. "Discovering Community Structure in Dynamic Social Networks using the Correlation Density Rank," *SocialCom - Stanford, CA, USA. The Sixth ASE International Conference on Social Computing*, 2014, http://www.ase360.org/handle/123456789/84.

[28] Baraniuk RG, Flandrin P, Hansen AJEM, Michel O (submitted):"Measuring time frequency information content using the Renyi entropies". *Postscript version available at www-dsp.rice.edu/publications.*

[29] Xing, W. and A. Ghorbani. "Weighted pagerank algorithm. in Communication Networks and Services Research", *Proceedings. Second Annual Conference on, IEEE*, 2004.

[30] Bahrami Bidoni, Z., George, R., & Shujaee, K. "A Generalization of the PageRank Algorithm." *ICDS 2014, The Eighth International Conference on Digital Society*, pp. 108-113. 2014.

[31] Anand, K. and G. Bianconi, "Entropy measures for networks: Toward an information theory of complex topologies". *Physical Review E,* 2009. 80(4): p. 045102.

[32] Hinneburg, A., & Keim, D. A. "An efficient approach to clustering in large multimedia databases with noise". *KDD,* 1998, Vol. 98, pp. 58-65).

[33] Kivimäki, I., M. Shimbo, and M. Saerens, "Developments in the theory of randomized shortest paths with a comparison of graph node distances". *Physica A: Statistical Mechanics and its Applications,* 2014. 393: p. 600-616.

[34] Saerens, M., et al., "Randomized shortest-path problems: Two related models". *Neural Computation*, 2009. 21(8): p. 2363-2404.

[35] Shannon CE, "A mathematical theory of communication. Part I", *Bell Sys Tech,* 1948, 27:379–423.

[36] Willians WJ, Brown ML, Hero AO, "Uncertainty, information and time-frequency distributions". *Proc SPIE Int Soc Opt Eng*, 1991, vol. 1566, p 144–156.

[37] Flandrin P, Baraniuk RG, Michel O, "Time frequency complexity and information". *Proc IEEE Int Conf Acoust, Speech, Signal Processing-ICASSP'94.* III, 1994, pp: 329–332.

[38] Gonzalez Andino, S. L., et al. "Measuring the complexity of time series: an application to neurophysiological signals." *Human brain mapping 11.1* , 2000, pp: 46-57.

[39] http://en.wikipedia.org/wiki/Normal_distribution.

[40] Ballı, S. and Korukoğlu, S., "Operating system selection using fuzzy AHP and TOPSIS methods," *Mathematical and Computational Applications*, 2009, Vol. 14, No. 2, pp. 119-130.

[41] Ertuğrul, İ. and Karakaşoğlu, N., "Comparison of fuzzy AHP and fuzzy TOPSIS methods for facility location selection," *The International Journal of Advanced Manufacturing Technology*, 2008, Vol. 39, No. 7-8,pp. 783-795.

[42] Lee, H. S. and Chou, M. T., "A fuzzy multiple criteria decision making model for airline competitiveness evaluation," Lecture Notes in Computer Science, 2006, No. 4252, pp. 902-909.

[43] Liang, G. S., "Fuzzy MCDM based on ideal and anti-ideal concepts," *European Journal of Operational Research*, 1999,.Vol. 112, No. 3, pp. 682-691.

[44] Wang, Y. J., Lee, H. S., and Lin, K., "Fuzzy TOPSIS for multi-criteria decision-making," *International Mathematical Journal*, 2003,.Vol. 3, No. 4, pp. 367-379.

[45] Wang, Y. J. and Lee, H. S., "Generalizing TOPSIS for fuzzy multiple-criteria group decision-making," *Computers and Mathematics with Applications,* 2007,.Vol. 53, No. 11, pp. 1762-1772.

[46] Hwang, C. L. and Yoon, K., "Multiple Attribute Decision Making: Methods and Application", *Springer, New York*, 1981.

[47] Zeleny, M., *Multiple Criteria Decision Making*, McGraw-Hill, New York, 1982.

Network Service Quality Rank: A Network Selection Algorithm for Heterogeneous Wireless Networks

Zeynab Bahrami Bidoni
Department of Computer and Information Systems
Clark Atlanta University
Atlanta, GA
z.bahrami62@gmail.com

Roy George
Department of Computer and Information Systems
Clark Atlanta University
Atlanta, GA
rgeorge@cau.edu

ABSTRACT
High-speed wireless services have achieved remarkable rates of growth in recent years. In order to survive in this competitive market, high levels of service performance are an effective way to improve customer satisfaction and loyalty. This paper aims to identify the best service provider in a heterogeneous wireless network so that we differentiate the quality of service (QoS) and provide a framework for analytical performance evaluation. This problem is considered a ranking problem in Multi-Criteria Decision Making, and so we formulize a novel method to compute collaboration performance utility for each provider. The compromise ranking technique (called VIKOR) is used to aggregate all utility values on alternatives and computes the best level of service among providers. The experimental evaluation results demonstrate the computational efficacy of the solution approaches and derive managerial insights.

General Terms
Heterogeneous Wireless Networks (HWNs), Quality of Service (QoS), Multi-Criteria Decision Making (MCDM).

Keywords
Network Service Quality Rank (NSQR); Correlation Density Rank (CDR); Diversity of Density (DOD).

1. INTRODUCTION
High-speed wireless service has achieved a remarkable market penetration in recent years with numerous service providers. Thus much effort has been concentrated on developing multi-criteria radio access technology (RAT) selection algorithms for heterogeneous wireless networks (HWNs) [1] based on criteria such as bandwidth, maximum data supported, security level provided, battery power consumption etc.

The contributions of this study are as follows: (1) Modeling the suppliers' performance comparison problem as a ranking problem in Multi-Criteria Decision Making (MCDM) [2], (2) Developing an approach to evaluate the probability of non-occurring negative events between two random users, while positively discriminating the events occurring between a user and its significant partners over those with less significant affiliates. (3) Computing the utility and efficiency of collaboration between each pair of alternative resources. Moreover, the tradeoff between criteria can be made by the VIKOR method [3].

ANCS '14, October 20–21, 2014, Los Angeles, CA, USA.
ACM 978-1-4503-2839-5/14/10.
http://dx.doi.org/10.1145/2658260.2661763

2. PROPOSED APPROACH
We consider this issue as a ranking problem in MCDM. Various attributes can be selected as criteria but, here, we focus on negative frequent events which may occur on interaction between users of any pair of device or service providers during a given big enough period of time, and effect on network's collaboration performance. In order to establish the decision matrix, scoring alternatives on criteria needed. So, we should assign to an alternative the performance efficiency value for collaboration with other alternatives. After that, we use VIKOR method which introduces the multi criteria ranking index based on the measure of "closeness" to the "ideal" solution. With the purpose of scoring alternatives on criteria, we proposed a new approach which has three main steps as follows which repeat for all failed type of occurrences:

Step 1: Arranging the failed frequency matrix of the heterogeneous network so that all users from same device or service provider set next to each other to extract sub-frequency matrices related to collaboration between each pairs of potential resource or device types.

Step 2: computing below items for all sub-frequency matrices;

- The Correlation Density Rank (CDR) vector [4] which is customized in this study to compute probability density of given events on homogenous network's users.

1. Initialize cost distance matrix C.
$$C[i,j] = \log_{(1-P_{ij}^{neg})}^{(1-\exp(-\gamma fij))} \quad (1)$$
(The logarithm of $(1-\exp(-\gamma f_{ij}))$ based on $(1-P_{ij}^{neg})$)
2. $M \leftarrow$ Normalize matrix C on columns.
3. For each node n_j ($1 < j < l$) compute inverse of the entropy of related column from matrix M:
$$ej \leftarrow -\frac{1}{Lnk}\sum_{i=1}^{k} M_{ij} Ln(M_{ij}) \quad (2)$$
$$\sigma j \leftarrow \frac{1}{e_j} \quad (3)$$
4. Calculate the density function which results from a Gauss Influence function.
$$cdr_i \leftarrow \sum_{j=1}^{l} \exp\left(-\frac{(C_{ij})^2}{2\sigma_j^2}\right) \quad (4)$$
5. Normalize Correlation Density Rank vector:
$$CDR_i \leftarrow cdr_i \Big/ \sum_{i=1}^{k} cdr_i \quad (5)$$
6. Return CDR.

- Unpredictability/ Diversity of Density (DOD) of failed occurrences. We employ the CDR vector as the probability density distribution in Renyi entropy formulate [5].

$$H_{kl}^{e_t} = \frac{1}{1-\alpha}\log_2\left(\frac{\sum_{i=1}^{N_k} CDR_i^{\alpha}}{\sum_{i=1}^{N_k} CDR_i}\right) \quad (6)$$

Where $H_{kl}^{e_t}$ the unpredictability of the failed event e_t between users from the network service provider or device type k to l, and N_k is the number of users from service provider or device type number K.

- The estimated probability of non-occurring any failed happening between two random users from given two resource or device types. Considering a Gaussian distribution with below mean and variance parameters, can help us to better understanding about probability distribution of given frequent event through kl cooperation network.

$$\mu_{kl}^{e_t} = \frac{\sum_{i=1}^{N_k}\sum_{j=1}^{N_l} F_{kl}(i,j)}{N_k N_l}, \quad \delta_{kl}^{e_t} = \frac{\mu_{kl}^{e_t}}{H_{kl}^{e_t}} \quad (7)$$

$$P_{kl}^{e_t} = p(x=0) = \frac{e^{\frac{-(\mu_{kl}^{e_t})^2}{2(\delta_{kl}^{e_t})^2}}}{\delta_{kl}^{e_t}\sqrt{2\pi}} \quad (8)$$

Step 3: scoring service providers or devices based on their interaction utility and efficiency with other alternatives as criteria, and constructing decision matrix to apply VIKOR method for ranking alternatives' performance. We follow a variant of the sigmoid functions to model the service user's satisfaction. The normalized user satisfaction is modeled as,

$$U_{kl}^{e_t} = \frac{1}{1+e^{\left(\frac{\left(P_{\max}^{e_t}+P_{\min}^{e_t}\right)}{2} - P_{kl}^{e_t}\right)}} \quad (9)$$

Where $U_{kl}^{e_t}$ is the user satisfaction perceived, and $P_{kl}^{e_t}$ is the probability of non-occurring which is computed by Eq. (8), for event e_t in interaction from alternative resource or device type k to l. $P_{\min}^{e_t}$ is the minimum and $P_{\max}^{e_t}$ is the maximum probability value through all k and l. To compute the efficiency of cooperation between two resource or device type, we proposed Eq. (10) to aggregate utilities of reciprocal interactions per pairs of options.

$$E_{(k,l)}^{e_t} = \frac{1}{1+\log_{(1-w_x)} U_{kl}^{e_t}} + \frac{1}{1+\log_{(1-w_y)} U_{lk}^{e_t}} \quad (10)$$

w_x and w_y are the importance weight of sending and receiving operation, respectively, where their summation is equal to 1.

This research is funded in part by the Army Research Laboratory under Grant No: W911NF-12-2-0067 and Army Research Office under Grant Number W911NF-11-1-0168. Any opinions, findings, conclusions or recommendations expressed here are those of the author(s) and do not necessarily reflect the views of the sponsor.

3. EXPERIMENT

To implement our approach, we consider a synthetic heterogeneous Communication Network with 20 Communication Modules, three alternative devices and two types of failed events during sample time period. Assume a new user wants to enter in this network and should select one of three types of devices. After employing the approach of scoring alternatives based on criteria, the decision matrix constructed is shown on Table 1.

Table 1. Decision matrix resulted by the proposed method

Criteria	Failure occurrence e_1			Failure occurrence e_2		
Sub-criteria	With A	with B	with C	with A	with B	with C
CModule A	0.73	1.80	0.571	1.99	1.85	1.91
CModule B	1.93	1.85	1.67	2	2	1.87
CModule C	0.64	2	1.90	1.89	1.81	2

Table 2. Results of ranking by the VIKOR method

Alternative	CModule A	CModule B	CModule C
Distance to Ideal	0.7439	0.32051	0.5

According to final results of VIKOR (Table 2.), CModule type B is closest one to ideal and then type C and A respectively are in descending orders as expected.

4. CONCLUSIONS

Ranking performance of resources in wireless networks has a broad range of applications such as comparison of resource or device performance in Computer Area, Telecommunication, Electrical, Social, Supply Chains, Financial Networks and etc. We establish the MCDM optimization model for selecting best resource based on their efficiency of collaboration with other alternatives about occurring negative frequencies. The proposed method is composed of three main parts: (1) evaluating the probability of non-happening the negative events between two random users which positively discriminate the events between a user and its important partners over those with less significant affiliates. (2) Computing the utility and efficiency of collaboration between each pair of alternative resources; and, (3) construct the decision matrix and employ the well-known VIKOR method to rank alternative resources. Finally, these algorithms were applied to synthetic Communication network. The results can automatically satisfy the requirements of QoS users preferentially.

5. REFERENCES

[1] Ernst, J. B., Kremer, S. C., & Rodrigues, J. J. (2014). A Survey of QoS/QoE mechanisms in heterogeneous wireless networks. *Physical Communication*.

[2] Triantaphyllou, E. (2000). *Multi-criteria decision making methods a comparative study*. Springer.

[3] Opricovic, S., & Tzeng, G. H. (2004). Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *European Journal of Operational Research,156*(2), 445-455.

[4] Bahrami Bidoni, Z., George, R. (2014). Discovering Community Structure in Dynamic Social Networks using the Correlation Density Rank. *SocialCom, The Sixth ASE International Conference on Social Computing.*

[5] Gonzalez Andino, S. L., et al. (2000). Measuring the complexity of time series: an application to neurophysiological signals. *Human brain mapping* 11(1), 46-57.

A Smart Assignment Technique with Consideration of Multicriteria Reciprocal Judgments

Zeynab Bahrami Bidoni; Roy George
Department of Computer and Information Systems
Clark Atlanta University
Atlanta, GA
z.bahrami62@gmail.com
rgeorge@cau.edu

Ahmad Makui
Department of Industrial Engineering
Iran University of Science and Technology (IUST)
Tehran, Iran
amakui@iust.ac.ir

***Abstract*— To date the assignment problems are important tasks in recommender systems and one-to-one matching issues through social environments. The various approaches have been proposed to reach these purposes that are normally limited to the considerations of cost or profit incurred by each possible assignment. However most of the time, each of the alternatives at both assignment sides have particular criteria for judging about the other side alternatives, whereby they can evaluate their sufficiency. In this paper, in order to obtain the optimality of both dimensions of assignment we try to consider the concept of efficiency rather than the cost or profit of each possible assignment. Therefore, the efficient assignment is the one that firstly, has the maximum optimality in terms of both dimensions of assignment, and secondly, takes into account the significance of judgment of each assignment from the viewpoint of decision maker. To do this, a compound index would be defined which includes the efficiency related to two-dimensional optimized assignment for the purpose of measuring the performance of each possible assignment. Next, A mathematical programming model for the extended assignment problem is proposed, which is then expressed as a classical integer linear programming model to determine the assignments with the maximum efficiency. A numerical example is used to demonstrate the approach.**

Keywords*— *Assignment Problem; Multicriteria Reciprocal Judgments; Two-dimensional Utility; Total Efficiency in Reciprocal Optimality; Virtual Alternative.

I. INTRODUCTION

The assignment problem is a common term in the theory of linear and network flow. This problem has been proposed in different forms [1] but it is most often considered in form of optimal solution of assigning 'n' jobs to 'n' people in a way that minimum cost or maximum profit would be obtained. You can see some of its usage in [2-6]; in order to find effective and optimal solutions, different algorithm including standard linear programming [7-12], Hungarian algorithm [13], neural network [14], and genetic algorithm [15-19] have been devised. For standard assignment problem, only the cost or the profit of each possible assignment are considered in formulation of the problem; but in real usage, for each possible assignment several types of input resources are usually needed in an assignment problem. Moreover, decision-makers can have several different objectives to achieve for each possible assignment, and the ways to achieve these objectives may conflict with each other. Cambell and Diaby in an article [20] pointed out that demand levels in different departments as well as the number of present workers should be regarded as the input, and the assignment outcomes can affect quality of service and employee satisfaction. They also emphasized that effective utilizing of human resources is of utmost significance in sensitive professions such as nursing.

Bera and Suer also claim that multiple factors can affect the assigning of human resources in the manufacturing cell. Overall, different evaluation units could be used to assess performance measurements of the objectives. These measurements are considered as the output of the problem. The problem can have several incompatible and opposing input and output. In this regard, in an article [22] the author has formulated a problem by considering multiple input and output for each possible assignment, and utilizes data envelopment analysis (DEA) for measuring the efficiency in proposed approach.

Chi-Jen Lin (2011), proposes a labeling algorithm to identify two other sensitivity ranges – Type II and Type III. The algorithm uses the reduced cost matrix, provided in the final results of most solution algorithms for AP, to determine the Type II range which reflects the stability of the current optimal assignment [23]. Birger Raa et al. (2011) In [24] present a MILP model for the integrated BAP–CAP taking into account vessel priorities, preferred berthing locations and handling time considerations. Robert F. Bordley & Stephen M. Pollock (2012) in [25] used an approach that maximizes organizational utility which is assumed to be zero if any of the activities cannot meet its target (or resource allocation). In their approach, utility-based probability maximization (UPM) is a variant of stochastic optimization without recourse.

The standard assignment problem is a particular form of the transportation problem and could be formulated in a linear integer programming of 1-0 [26-27], as follows:

ISBN: 978-1-62561-003-4

$$Min\ (or\ Max) \quad \phi = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}s_{ij}$$

Subject to

$$\sum_{j=1}^{n} s_{ij} = 1\ ,\ \ i = 1,\dots,n$$
$$\sum_{i=1}^{n} s_{ij} = 1\ ,\ \ j = 1,\dots,n \qquad (1)$$
$$s_{ij} = 0\ ,or\ 1$$

In which the decision variable $s_{ij} = 1$ means that 'i' th individual is assigned to 'j' th job, while for $s_{ij} = 0$ no assignment is made. c_{ij} is the cost (or profit) imposed by the assignment. Particular computer software could easily be used to solve above formulated problems as well as to find the set of optimal answers for identifying the minimum cost and maximum profit. But it should be noted that in this formulation, the cost or profit is only regarded for measuring the function and as we mentioned earlier, other criteria rather than profit or cost could be used for measuring the function of assignments.

The basic idea of performing this research has been derived from the assignment problem which encounters in real positions and is not solvable with current methods. The problem is that we want to optimally assign some of employees to some jobs in a way that each of occupations needs some kind of capability and eligibility as evaluation criterion. Meanwhile, manager as decision maker in order to enhance sense of job satisfaction wants to take into account tastes and utility of employees in case of each job. Meanwhile, imposing each person's taste and also qualifications and capabilities needed for every job have different level of importance. Therefore, we deal with assignment problem of two goals: first, to maximize degree of utility in view of each person's taste and second, to maximize degree of utility from the dimension of qualification and competency needed for each occupation according to the priority of each items. As another example we can consider a coach as a decision maker who intends to divide his/her students into different teams in different sports with limited space; in this decision making process he should take into account the qualifications and capabilities required for each sport area as well as the taste of the individuals so that the teams could have the required conditions for success.

Therefore, in this study, the maximum of the total efficiency in obtaining the optimality of both dimensions of assignments would be considered as the criterion of optimal assignment according which this study is organized and you could see what will come next in this paper. Part two would discuss about the overall structure of the model and would provide a definition of the problem. Part three put forward an approach for solving the problem and finding the optimal answer. Part four presents an example to better explain the approach, and finally part five deals with the conclusion of the study.

II. THE OVERALL STRUCTURE OF PROBLEM

Among the basic concepts required for elaborating the model of the problem, are the three concepts of 'alternative role', 'arbiter role' and 'decision maker role'. When an element has the role of an arbiter, it means that it has some criteria for measurement and can assess and order the opposite alternatives. The element that is being judged has the role of an alternative. The element which directly utilizes the assessments and judgments to the final solution has the role of a decision maker. Therefore, the element that has the role of a decision maker has definitely the role of an arbiter, but the element with the role of an arbiter does not necessarily have the role of a decision maker.

Here, we consider the decision making system as consisting of three distinct types of elements (a component of the decision making system called "element" that could accept one or more role of the tree role of "alternative", "arbiter" or "decision maker"). Both the elements of X, Y have the roles of 'alternative' and 'arbiter' reciprocally, and the third element, that is Decision Maker (DM), has the role of a decision maker which is the one responsible for doing the assignment task (See Fig .1). We assume to have 'k' elements of the 'x' type, each of them are shown as X_i, $i = 1,2,\dots,k$; on the other side we have 'l' elements of the 'y' type that each of them are shown as Y_j, $j = 1,2,\dots,l$ $(k \leq l)$. $C^X = \{c_1^X, c_2^X, \dots, c_r^X\}$ is the set of the references of the attributes related to the assessment of Xs and $C^Y = \{c_1^Y, c_2^Y, \dots, c_s^Y\}$ is the set of the references of attributes related to the assessment of Ys. In this problem each element X_i, $i = 1,2,\dots,k$ takes into account some attributes of C^Y as the criterion of assessment and judgment about all $Y_j s$, and also each of Y_j, $j = 1,2,\dots,l$ has considered a subset of C^X attributes for the sake of measurement and judgment about all X_is. Now DM is the one that makes decision about the assignment of elements of the Y type to the elements of the X type and intends to perform the assignment in a way that the maximum optimality is obtained observing the criteria of the elements of the both sides. It should be noted that each element Y_j could only be assigned to one element X_i and the assignment capacity for each X_i, $i = 1,2,\dots,k$ equals the number of P_i (P_i is Natural number and $\sum_{i=1}^{k} P_i \leq l$).

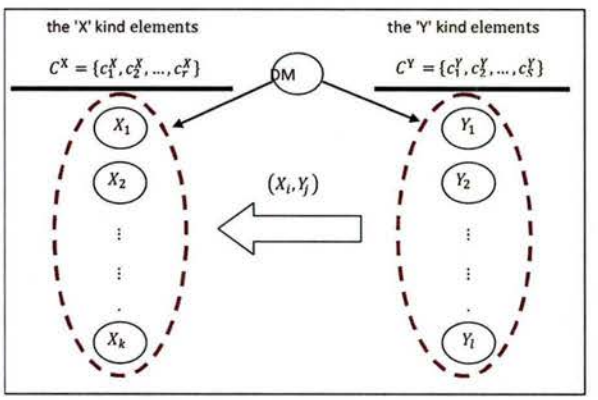


Fig. 1. The structure of assignment model based on multicriteria reciprocal judgments

In addition, DM may attach different significance to judgment of the elements X and Y, therefore, W_X is the significance weight of the elements of the X type and W_Y is the significance weight of elements of the Y type and accordingly $W_X + W_Y = 1$. Also among the elements of X type, the DM may attach different importance to X_is in which case w_{x_i}, $i = 1,2,\dots,k$ is the significance weight of element

X_i in terms of DM in a way that $\sum_{i=1}^{k} w_{x_i} = 1$. In the same vein, about elements of Y type w_{Y_j} , $j = 1,2,\dots,l$ is the significance weight of element Y_j in terms of DM so that $\sum_{i=1}^{l} w_{Y_j} = 1$.

Because of the reciprocity of alternative role in this decision making model, the set of alternatives to be considered in this problem is a two dimensional set which can be viewed as a set of virtual alternatives that are ordered pairs and each of their component is related to each side of the assignment. Therefore the set of $X \times Y$ would be the set of the alternatives to be considered and is represented as follows:

$$A \stackrel{\text{def}}{=} X \times Y = \{(X_i, Y_j) | X_i \epsilon X , Y_j \epsilon Y\} \quad (2)$$

Each alternatives of (X_i, Y_j) from the set of A $(i = 1,2,\dots,k \ , \ = 1,2,\dots,l \ )$ is interpreted as the assignment of element Y_j to the element X_i . With such a definition, we are dealing with a problem of multi-criteria decision analysis which has $l \times k$ alternatives and one decision maker (DM). It should be noted that each of the X_i, $i = 1,2,\dots,k$ and each of the Y_j , $j = 1,2,\dots,l$ could be arbiter only about alternatives in which one of their components is included. In order to simplify the issue, we define some restrictions of set of A as follows:

$$\begin{aligned} \forall_{i=1,2,\dots,k} \ A_{x_i} &:= \{(X_i, Y_j) | Y_j \epsilon Y\} \quad , \\ \forall_{j=1,2,\dots,l} \ A_{y_j} &:= \{(X_i, Y_j) | X_i \epsilon X\} \end{aligned} \quad (3)$$

So with this definition we can say that each element X_i , $i = 1,2,\dots,k$ has the role of arbiter only toward the virtual alternatives of set A_{x_i}, as well as each of the element Y_j , $j = 1,2,\dots,l$ only toward the virtual alternatives of set A_{Y_j}; but DM is the element that has the role of arbiter and decision maker toward all elements of set A. Now, we try to find an algorithm to solve the problem whereby we could obtain the best assignment with maximum optimality in terms of elements of X and Y type.

III. Problem Solving Approach

Here we are dealing with an assignment problem in which decision maker intends to process the assignment in a way that the maximum optimality could be obtained in terms of both sides of the assignment. Regarding this, first the procedure of ranking which is frequently used in this algorithm would be defined and notated.

A. The ranking procedure:

The purpose of utilizing the ranking procedure is to recognize the criteria, value functions and the mental ideal point of the decision maker on the criteria and to rank the alternatives by measuring the preferable distance of each alternative from the ideal point, so that in terms of the preference amount, the closest alternative to the ideal point would gain the first rank, and in the same way, the remaining alternatives would obtain the next ranks. The symbol of this procedure is written as $\text{rank}_*(*,*)$. As an example we could assume that the element b is the arbiter and the set $A = \{a_1, a_2, \dots, a_g\}$ is the set of to-be-considered alternatives. The set $U = \{u_1, u_2, \dots, u_q\}$ is also the reference set of the criteria. Therefore, $\text{rank}_b(A, U)$ is the ranking of the set of alternatives A by the arbiter b which is based on the arbitrary criterion of the arbiter among the criteria of the reference set U which is done through these procedures:

Step1. Choosing the criteria: the arbiter would be asked to choose a subset of arbitrary criteria based on which he wishes to do the ranking from the reference set U; the set of chosen criteria is called C.

$$C = \{c_1, c_2, \dots, c_n\} \subseteq U \ \ ; \ \ |C| = n \quad (4)$$

Step2. Giving weight to the chosen criteria: in this step we can directly ask the arbiter to provide us with the weight of the criteria and if not possible we can calculate the weights of the criteria through one of the common ways of weight-giving to match in the following conditions.

$$\sum_{i=1}^{n} w_i = 1 \ , \ w_i > 0 \quad ; \quad W = (w_1, w_2, \dots, w_n) \quad (5)$$

Step3. Identifying the value function related to each criterion by the arbiter: in this phase the arbiter would be asked to identify the mental value function in respect to each criterion. In these functions, the horizontal axe represents the value of outcomes in intended criterion, and the vertical axe is related to the value that those outcomes have for the arbiter. Here, we define 3 aspiration levels for the value size and we ask the arbiter to identify the value size related to outcomes of each criterion based on these levels. These levels are as follows: 1. "quite dissatisfaction" which has the zero value. 2. "quite satisfaction" which has the value of one. 3. "quite surprised " which has the value of two.

If the outcome of a criterion is quite satisfactory for the arbiter, we give value 1 to that outcome in the vertical axe, and in the same vein, for each outcome based on the relative satisfaction it creates for the arbiter, we assign values equal, smaller or larger than 1. The smaller the value is than 1, the more arbiter would be dissatisfaction; and the more it is than 1, the arbiter would be more Surprised. In fact the range of the value function would be between zero to two in which 1 indicates the quite satisfaction and 1 to 2 represents that arbiter is Surprised. As an example, the value function could be as follows:

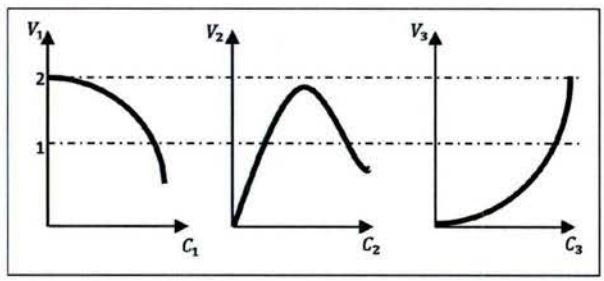


Fig. 2. Three instances of identified value function by the arbiter.

The value function of v_i is a converter that transforms the outcome value obtained from the alternatives a_j, $j = 1,2,\dots,g$ on the c_i criterion to the value defined by the arbiter.

$$v_i^j = F_i\left(C_i(a_j)\right) = v_i(a_j) \ \ ; \ \ i = 1,2,\dots,n \ \ ; \ \ j = 1,2,\dots,g \quad (6)$$

We define the vector 'V' that has 'n' components as the vector of value functions and in each of its component we put the value function related to one of these criteria.

$$V^j = \left(v_1^j, v_2^j, \dots, v_n^j\right) = \left(v_1(a_j), v_2(a_j), \dots, v_n(a_j)\right); \quad j = 1,2,\dots,g \tag{7}$$

Note: in the aggregation model, in order to obtain the whole preferences of the arbiter or decision maker on the alternatives, the assessment criterion for each alternative is considered as a function of value functions and the vector of criteria weight and based on the values obtained from this aggregation function would be ranked in descending order that is the alternative that gain the highest value in the aggregation function would get the rank 1 and the others would be ranked based on the same vein. But get the aggregation function is very difficult because autonomy and dependency status should be among the criteria considered. Sometimes, considering all these relations will not be practical. But this model is proposed a method that to obtain ranking and of aggregation function is not used.

Step4. Formation of n-dimension space with value functions and identification of each alternative a_j $(j = 1,2,\dots,g)$ as a point with the coordinates of V^j: we assume to show each of the alternatives of $A = \{a_1, a_2, \dots, a_g\}$ with n-component vector so that the i' th component related to a_j $(i = 1,2,\dots,n\,;\, j = 1,2,\dots,g)$ is the outcome of alternative a_j in the i' th criterion. In this way we could consider the alternatives as points in n-dimension space of criteria. Therefore, each alternative could be displayed with his value functions vector that is alternatives could be considered as points in the n-dimension space of value functions. As an exemplary assumption take n=3 that mean we have 3 criteria so the 3 dimension of criteria and the 3-dimension space of value functions is as follows:

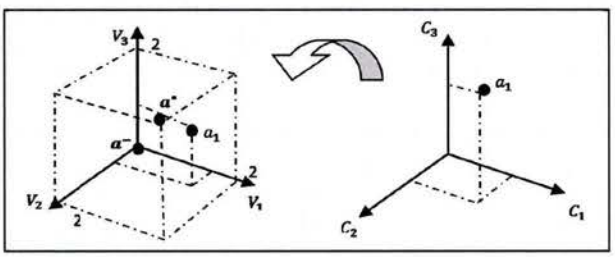


Fig. 3. Definition of the alternatives as the points in the space of criteria and its transference to the value functions space.

Therefore, we would consequently have n-dimension space that each of its dimensions is the identifier of value function related to one of the n-criterion of the arbiter; and each alternative a_j $(j = 1,2,\dots,g)$ in this space has the n-component coordinates that could be considered as a spot (point) of this space in a way that the i' th component of each coordinate of alternative equals the value, the outcome of which is obtained in terms of the i' th criterion of arbiter 'b'. The point to be noted is that the value functions space and the coordinates of an alternative in this space is strictly dependent on the idealizations of the arbiter 'b', since the criteria are selected by the arbiter as well as identification of value functions. Therefore, a particular alternative may have quite different coordinates in the space of arbiters' value functions either in terms of the number of the components or in terms of the value of each of the coordinate's components. Consequently we can explain the n-dimension value functions space of arbiter 'b' as the "n-dimensional space of 'b' idealization".

Since the range of the value functions of v_i, $i = 1,2,\dots,n$ is between zero and 2, the ideal point could be considered as a point of the value functions space in which all the components of coordinate is 2, that is the alternative that has obtained the highest value in view of the arbiter and would be represented as $a^* = (2,2,\dots,2)$. In the similar vein the negative ideal point is the one that has obtained the lowest value in view of the arbiter, so all the components of the coordinate equals zero and is displayed as $a^- = (0,0,\dots,0)$. it should be noted that, here we assume a^- has never been a member of the alternatives to be considered, for the simple reason that the occurrence of such a phenomenon i.e., existence of such an alternative that in view of all criteria has the absolute zero value is quite rare and almost impossible. If by chance such an alternative exists, it could be removed from the set of to-be-considered alternatives from the very outset.

Step5. Calculating the closeness of relational preference of alternatives to the ideal point and their ranking based on this index: in this step, we obtain the Euclid distance, between the identifier point of each alternative in value functions space, from the two ideal point and negative ideal point as follows:

$$S_j^* = \sqrt{\sum_{i=1}^{n} w_i * \left(2 - v_i^j\right)^2}\;; \quad j = 1,2,\dots,g \tag{8}$$

$$S_j^- = \sqrt{\sum_{i=1}^{n} w_i * \left(v_i^j\right)^2}; \quad j = 1,2,\dots,g \tag{9}$$

In which for S_j^*, $j = 1,2,\dots,g$ consists of the Euclid distance of alternative a_j from the ideal point and S_j^-, the Euclid distance of alternative a_j from the ideal negative point in the value functions space. Now, in order to rank the set of alternatives A, we define an index termed as "closeness of relational preference to the ideal point" as what you could see below:

$$RPC_b^{a_j} = \frac{S_j^-}{S_j^* + S_j^-} \quad ; \; j = 1,2,\dots,g \tag{10}$$

The $RPC_b^{a_j}$ is the indicator of the closeness of relational preference of alternative a_j to the ideal point of a^* in idealization space of the arbiter b. if $a_j = a^*$, $RPC_b^{a_j}$ equals 1 and when $a_j = a^-$ it equals zero, but since we assume that a^- is not a member of to-be-considered alternatives of A, always we have $0 < RPC_b^{a_j} \leq 1$ $(j = 1,2,\dots,g)$. the higher is the index for one alternative, the closer the alternative is to the ideal in terms of the preferences of arbiter element 'b', and at the same time it is farther from the negative deal. Finally we order and rank the alternatives of set A, in descending order, from the highest proximity of relational preference to the ideal point, to its lowest proximity.

Now for solving this problem and obtaining the most appropriate assignment, we suggest the following phases:

Phase 1: we utilize the ranking procedure for every single elements of X and Y type in arbiter position:

$$\forall_{X_i, i=1,\dots,k} \quad \text{rank}_{X_i}(Y, C^Y) \quad , \quad \forall_{Y_j, j=1,\dots,l} \quad \text{rank}_{Y_j}(X, C^X) \tag{11}$$

Phase 2: in this phase we form the decision matrix of problem by applying the result obtained from the first phase as follows. As we know, we are dealing with a reciprocal judgment and each assignment of a Y type element to a X type element form a to-be-considered alternative which are shown as (X_i, Y_j), $i = 1,2,\dots,k$ and $j = 1,2,\dots,l$. In addition, we want to assess each of these alternatives as tow attributes the first attribute (U^X) is the amount of relative utility of this assignment in terms of X type element, and the second attribute (U^Y) is the amount of the relative utility of this assignment in terms of Y type element. We show the set of these two attributes as $U^{XY} = \{U^X, U^Y\}$. Therefore, the decision matrix structure would be defined as follows:

A \ U^{XY}	U^X	U^Y
(X_1, Y_1)	U_{11}^X	U_{11}^Y
⋮	⋮	⋮
(X_i, Y_j)	U_{ij}^X	U_{ij}^Y
⋮	⋮	⋮
(X_k, Y_l)	U_{kl}^X	U_{kl}^Y

Fig. 4. The structure of decision matrix in assignment model based on the multi-criteria reciprocal judgments.

In which the U_{ij}^X ($i = 1,2,\dots,k$ and $j = 1,2,\dots,l$) is the result or the outcome of the judgment of X type element about the assignment (X_i, Y_j) and equals the amount of relative utility of element Y_j in terms of element X_i, also U_{ij}^Y, ($i = 1,2,\dots,k$ and $j = 1,2,\dots,l$) is the result or outcome of the judgment of Y type element about the assignment (X_i, Y_j) and equals the amount of relative utility of element X_i in view of element Y_j. Now the question is that how are these outcomes obtained? For each $i = 1,2,\dots,k$ and $j = 1,2,\dots,l$ we define the outcomes of decision matrix in following way:

$$U_{ij}^X := w_{X_i} \cdot RPC_{X_i}^{Y_j} \quad ; \quad U_{ij}^Y := w_{Y_j} \cdot RPC_{Y_j}^{X_i} \tag{12}$$

So for each $i = 1,2,\dots,k$ and $j = 1,2,\dots,l$ We have $0 < U_{ij}^X, U_{ij}^Y \leq 1$.

Phase 3: Now we define an index that could be used as the decision criteria in solving problem. This index is called "total efficiency in reciprocal optimality" and for each alternative (X_i, Y_j), $i = 1,2,\dots,k$ and $j = 1,2,\dots,l$,we show it with E_{ij}. We could consider this index as a linear combination of U_{ij}^X, U_{ij}^Y, that is if the decision maker (DM) give the weight W_X to the X type element judgment and give W_Y to the judgment of Y type element so that $W_X + W_Y = 1$, $0 < W_X, W_Y < 1$, then the linear combination of $E_{ij} := W_X \cdot U_{ij}^X + W_Y \cdot U_{ij}^Y$ could be considered as an index for measuring the "total efficiency in reciprocal optimality". However, the point to be noted is that in this definition, the E_{ij} derivative in relation to U_{ij}^X or U_{ij}^Y is a constant value, which is the ratio of the total efficiency changes to the relative utility changes of the alternative is a constant value. While, commonly the closer the amount of relative utility is to 1, and the alternative has higher level of satisfaction, the less the sensitivity would be toward the optimality changes. For this reason, we should define index E_{ij} in a way that it owns this characteristic. Here we define the index E_{ij} for the alternative (X_i, Y_j), $i = 1,2,\dots,k$ and $j = 1,2,\dots,l$ in following way:

$$E_{ij} := \frac{1}{1 + \log_{W_Y} U_{ij}^X} + \frac{1}{1 + \log_{W_X} U_{ij}^Y} \tag{13}$$

In which the first sentence shows the efficiency of (X_i, Y_j) in X dimension and the second sentence shows the efficiency in Y dimension. Also it is clear that $0 < E_{ij} \leq 2$.

The reasons that confirm the appropriateness of the above definition for E_{ij}:

1. Function $\frac{1}{1+\log_b X}$ (0<b<1) is shown in picture5. Simply we could see that the derivative of this function is positive (ascendant) and its second derivative is negative. In fact the more we get closer from x=1 to x=0 the slope of the graph slowly become lower which is compatible with what is in the mind of the decision maker. Because the more the relative utility is and closer to 1, the less the sensitivity of the decision maker is toward the optimality changes, in other words when the relative utility of an alternative gets higher the speed of the efficiency changes becomes lower.
2. This definition maintains the order and density of the preferences properly, and calculates the efficiency size with regard to the significance weight related to the judgment of each dimension based on the relational preferences.
   - Without disturbing the totality of the problem, we consider the statement related to the efficiency from the dimension of X ($\frac{1}{1+\log_{W_Y} U_{ij}^X}$), if we consider the sentence related to the efficiency from the Y dimension the change procedure would be the same. Therefore, the consideration of one of these two is sufficient.
   - We assume U_{ij}^X to be constant and increase the W_Y; consequently, as shown in Fig.5, the value of $\frac{1}{1+\log_{W_Y} U_{ij}^X}$ would decrease; in fact the efficiency would decrease from the X dimension. In other word, if we assume the amount of relative utility of assumed alternative (assignment of Y_j to X_i) in view of X_i judgment as constant and increase the amount of W_Y, in fact we have decreased the significance of judgment in X dimension, because $W_X = 1 - W_Y$ and it is normal that the efficiency get decreased in X dimension. Also if we decrease W_Y, consequently, the W_X get increased and the value of $\frac{1}{1+\log_{W_Y} U_{ij}^X}$ would increase and in fact the efficiency would increase from x dimension.

- Now, we keep W_Y as a constant and distinct value and increase the amount of U_{ij}^X. As in Fig. 5, we will see that $\frac{1}{1+\log_{W_Y} U_{ij}^X}$ increases in parallel with increase in U_{ij}^X. That is if we assume W_Y as constant value (in fact, the significance weight related to both dimension is assumed to be constant and specified), based on the graph in Fig. 5, the more the value of relative utility of alternative is in view of X_i judgment, the more the efficiency value would get from the X dimension. The reverse also turns out to be true, that is the decrease of U_{ij}^X for one alternative leads to decrease its efficiency in view of X dimension. However it should be noted that the slope of efficiency changes would decrease with an increase in relative utility value and it is exactly what happens in the decision maker mind, because with increase in the relative utility level, the sensitivity of the decision maker about these changes would decrease and this feature is well considered in the definition of efficiency.
- If U_{ij}^X and U_{ij}^Y have the same value and $W_X > W_Y$, then based on the assumed definition, the efficiency of the alternative from X dimension would get higher than the efficiency of the alternative from the Y dimension.
- If $W_X = W_Y$, meaning that the significance of judgment of X and Y dimension is the same in decision making and $U_{ij}^X > U_{ij}^Y$, then the based on the definition, the efficiency from X dimension would be higher than the efficiency from the Y dimension.

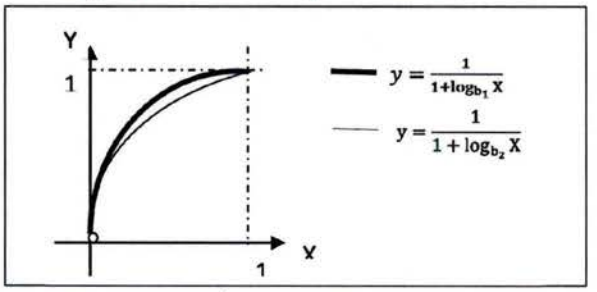


Fig. 5. *The graph of* $Y=\frac{1}{1+\log_b X}$ *when* $0 < b_1 < b_2 < 1$

It should be noted that, here, we aren't looking for the numerical value of efficiency index, but what is important is that this index could properly identify the total preference order based on the two dimension optimality of the to-be-considered alternatives, and since the behavior of defined formulation for E_{ij} is all the time in consistent with reality of decision maker mentality, it seems that this definition is more appropriate and efficient than he basic definition (linear compound). So in this phase for each assignment(X_i, Y_j), $i = 1,2,\dots,k$ and $= 1,2,\dots,l$, the numerical value of E_{ij} would be calculated based on the defined formulation.

Phase 4: In target function of assignment problem, in order to identify the maximum of 2-dimension optimality measurement, the values of E_{ij} would be utilized and the problem would be formulated in following way:

$$MAX \quad \phi = \prod_{i=1}^{k}\prod_{j=1}^{l} E_{ij}^{s_{ij}}$$

Subject to :

$$\sum_{j=1}^{l} s_{ij} = P_i \quad ,i = 1,2,\dots,k \qquad (14)$$

$$\sum_{i=1}^{k} s_{ij} \le 1 \quad ,j = 1,2,\dots,l$$

$$s_{ij} = 0 \text{ or } 1$$

In which the $s_{ij} = 1$, that is the assignment of element Y_j to the element X_i is done through DM while for $s_{ij} = 0$ no assignment has taken place. Since $0 < E_{ij} \le 2$ and $s_{ij} = 0$ or 1, we could conclude that $0 < \phi$. Therefore, by calculating the logarithme from the target function of ϕ, we could convert the above non-linear problem to the following linear programming problem.

$$MAX \quad \psi = \log\phi = \sum_{i=1}^{k}\sum_{j=1}^{l} s_{ij} \cdot \log E_{ij}$$

Subject to :

$$\sum_{j=1}^{l} s_{ij} = P_i \quad ,i = 1,2,\dots,k \qquad (15)$$

$$\sum_{i=1}^{k} s_{ij} \le 1 \quad ,j = 1,2,\dots,l$$

$$s_{ij} = 0 \text{ or } 1$$

The above linear programming would certainly have one optimal answer, and that answer would indicate the optimal assignment of the set of Y type element to the set of X type element by DM and with considering the utility of both decision dimensions.

IV. Numerical Example

Here we provide a real experiment about the assignment of job positions to individuals in order to demonstrate the applicability of the suggested approach in which a corporation manager, as a decision maker, requires the decision analysis techniques for assigning three employees (Y_1, Y_2, Y_3) to two jobs of store management (X_1) and finance manager (X_2) so that by considering the utility and employees' interest and also the required competencies of each job, decide on the best assignment in a way that the utility of both sides is being satisfied as far as possible. Here each job is being occupied by one person and the weight factor which the manager allocates for assignment system element is as follows:

$$W_X = 0.7 \quad , \quad W_Y = 0.3$$
$$W_{Y_1} = 0.4 \quad , \quad W_{Y_2} = 0.4 \quad , \quad W_{Y_3} = 0.2$$
$$W_{X_1} = 0.6 \quad , \quad W_{X_2} = 0.4$$

In order to extract the required data, we have designed some simple question forms, which were customized to estimate alternative's score on each criterion. In these forms, the person had selected a number in range 1 to 5 for each question to demonstrate his/her preference, and total average number was final score, which is converted in range o to 1.

TABLE I. INTRODUCTION OF THE SET OF CRITERIA C^X

C^X	C_1^X	C_2^X	C_3^X	C_4^X
criteria	Salary and benefits	Responsibility amount	Nature of work	Popularity in that department

TABLE II. INTRODUCTION OF THE SET OF CRITERIA C^Y

C^Y	C_1^Y	C_2^Y	C_3^Y	C_4^Y
criteria	Management power	Job-related education	Job-related experience	Public Relations

TABLE III. WEIGHT-GIVING TO CRITERIA FOR JUDGMENT ABOUT YS IN VIEW OF XS

	C_1^Y	C_2^Y	C_3^Y	C_4^Y
X_1	0.4	0.2	0.3	0.1
X_2	0.4	0.1	0.4	0.1

TABLE IV. WEIGHT-GIVING TO CRITERIA FOR JUDGMENT ABOUT XS IN VIEW OF YS

	C_1^X	C_2^X	C_3^X	C_4^X
Y_1	0.5	0.3	0.2	-
Y_2	0.4	0.2	0.2	0.1
Y_3	0.3	-	0.2	0.5

TABLE V. THE VALUES OF Y_iS ON CRITERIA IN VIEW OF X_1

	C_1^Y	C_2^Y	C_3^Y	C_4^Y
Y_1	1	0.7	0.8	0.1
Y_2	0.2	0.3	0.2	0.1
Y_3	0.2	0.4	0.3	0.5

TABLE VI. THE VALUES OF Y_iS ON CRITERIA IN VIEW OF X_2

	C_1^Y	C_2^Y	C_3^Y	C_4^Y
Y_1	1	0.2	0.4	0.1
Y_2	0.2	0.5	1	0.1
Y_3	0.2	0.4	0.5	0.5

TABLE VII. THE VALUES OF X_iS ON CRITERIA IN VIEW OF Y_1

	C_1^X	C_2^X	C_3^X
X_1	0.8	0.4	0.7
X_2	1	0.3	0.3

TABLE VIII. THE VALUES OF X_iS ON CRITERIA IN VIEW OF Y_2

	C_1^X	C_2^X	C_3^X	C_4^X
X_1	0.7	0.3	0.2	0.1
X_2	1.2	0.6	1	0.4

TABLE IX. THE VALUES OF X_iS ON CRITERIA IN VIEW OF Y_3

	C_1^X	C_3^X	C_4^X
X_1	0.4	0.2	0.3
X_2	0.7	1	0.7

TABLE X. DECISION MATRIX

U^{XY} / A	U^X	U^Y
(X_1, Y_1)	0.223	0.134
(X_1, Y_2)	0.062	0.096
(X_1, Y_3)	0.121	0.032
(X_2, Y_1)	0.128	0.138
(X_2, Y_2)	0.122	0.188
(X_2, Y_3)	0.078	0.076

TABLE XI. THE TOTAL EFFICIENCY IN RECIPROCAL OPTIMALITY FOR EACH POSSIBLE ASSIGNMENT

Index / A	E_{ij}	$\log E_{ij}$
(X_1, Y_1)	0.596	- 0.225
(X_1, Y_2)	0.434	- 0.362
(X_1, Y_3)	0.457	- 0.34
(X_2, Y_1)	0.522	- 0.282
(X_2, Y_2)	0.54	- 0.268
(X_2, Y_3)	0.442	- 0.354

At this stage, we solve the following linear programming to achieve the assignment with maximum efficiency.

$$MAX \quad \psi = \log \phi = -0.225\, S_{11} - 0.362\, S_{12} - 0.34\, S_{13} - 0.282\, S_{21} - 0.268\, S_{22} - 0.354\, S_{23}$$

Subject to:

$$S_{11} + S_{12} + S_{13} = 1$$
$$S_{21} + S_{22} + S_{23} = 1$$
$$S_{11} + S_{21} \le 1$$
$$S_{12} + S_{22} \le 1$$
$$S_{13} + S_{23} \le 1$$
$$S_{ij} = 0 \text{ or } 1$$

The optimal answer of this linear programming is: $S^* = (1,0,0,0,1,0)$. That is only amount of S_{11} and S_{22} are 1 and This means that (According to the proposed method in this study) assigning individual Y_1 to position of store management and individual Y_2 to position of finance manager were appropriate decision With regard to judgments of two fronts of assignment. In practice, after assigning new managers based on obtain results. The satisfaction survey (by question forms) demonstrates satisfaction in these two departments over %75 increased than before on both upper managers and employees' levels.

CONCLUSIONS

In this study with proposing a novel viewpoint on the basis of existing reciprocal system of judgment between the alternatives of the both side of assignment, the objective would be to maximize the assignment efficiency in obtaining the two dimension optimality with which cost and profit gets substituted which was considered in standard assignment problem, and for this purpose, a compound index was defined for measuring the function of each possible assignment in problem formulation. Then a mathematical programming model was proposed for problem solution and for determining the assignment with maximum efficiency it was transformed to a classic linear programming model.

REFERENCES

[1] Pentico David W. Discrete Optimization Assignment problems: A golden anniversary survey. European Journal of Operational Research 2007; 176:774–793.

[2] Soumis F, Ferland j, Rousseau J. A model for large-scale aircraft routing and scheduling problems. Transport. Res. Part B: Meth 1980;14(1):191–201.

[3] Campell J.F, Langevin A.The snow disposal assignment problem. J. Oper. Res. Soc 1995; 48: 919–929.

[4] Leblanc L.J, Farhangian K. Efficient algorithm for solving elastic demand traffic assignment problem and mode split-assignment problem. Transport. Sci 1981; 15 (4): 306–317.

[5] Dessouky M.M, Kijowski B.A. Production scheduling on single-stage multiproduct batch chemical process with fixed batch sized. IIE Trans 1997; 29 (5): 399–408.

[6] Mckeown P, Workman B. A study in using linear programming to assign students to schools. Interfaces 1976; 6 (4): 101–96.

[7] McGinnis L.F. Implementation and testing of a primal-dual algorithm for the assignment problem. Oper. Res. 1983; 31(2):277-291.

[8] Balinski M.L. A competitive (dual) simplex method for the assignment problem. Math. Program 1986;34 (2) :125–141.

[9] Hung M.S, Rom W.O. Solving the assignment problem by relaxation. Oper. Res. 1980;28 (4): 969–982.

[10] Barr R.S, Glover F, Klingman D. The alternating basis algorithm for assignment problems. Math. Program 1977;13 (1) : 13–1.

[11] Volgenant A. Discrete Optimization Solving the k-cardinality assignment problem by transformation. European Journal of Operational Research 2004;157: 322–331.

[12] Deng Kui Huang, Huan Neng Chiu, Ruey Huei Yeh, Jen Huei Chang, A fuzzy multi-criteria decision making approach for solving a bi-objective personnel assignment problem. Computers & Industrial Engineering ;2008.

[13] Kuth H.W. The Hungarian method for the assignment problem. Nav. Res. Log. 1955; 2: 83–97.

[14] Eberhardt S.P, Duad T, Kerns A, Brown T.X, Thakoor A.P. Competitive neural architecture for hardware solution to the assignment problem. Neural Networks. 1991; 4(4):431–442.

[15] Gunawan Aldy, Ng K.M, Ong H. L. A Genetic Algorithm for the Teacher Assignment Problem for a University in Indonesia. Information and Management Sciences. 2008; 19(1): 16-1.

[16] Anshuman Sahu, Rudrajit Tapadar, Solving the Assignment problem using Genetic Algorithm and Simulated Annealing. IAENG International Journal of Applied Mathematics. 36:1, IJAM_36_1_7.

[17] Avis D, Devroye L. An analysis of a decomposition heuristic for the assignment problem. Oper. Res. Lett. 1985; 3 (6): 279–283.

[18] Linzhong Liu, Xin Gao, Fuzzy weighted equilibrium multi-job assignment problem and genetic algorithm. Applied Mathematical Modelling 2009; 33: 3926–3935.

[19] Ahuja R.K, Orlin J.B, Tiwari A. A greedy genetic algorithm for the quadratic assignment Problem. Computers & Operations Research 2000; 27: 917-934.

[20] Campbell G.M, Diaby M. Development and evaluation of an assignment heuristic for allocating cross-trained workers. Eur. J. Oper. Res. 2002;138 :12–9.

[21] Su¨er G.A, Bera I.S. Optimal operator assignment and cell loading when lot-splitting is allowed. Comput. Ind. Eng. 1998; 35 (3–4) : 431–434.

[22] Chen L-H, Lu H-W. An extended assignment problem considering multiple inputs and outputs. Applied Mathematical Modelling 2007;31: 2239–2248.

[23] Chi-Jen Lin. A labeling algorithm for the sensitivity ranges of the assignment problem. Applied Mathematical Modelling. October 2011; 35(10): 4852–4864.

[24] Birger Raa, Wout Dullaert, Rowan Van Schaeren. An enriched model for the integrated berth allocation and quay crane assignment problem. Expert Systems with Applications. October 2011;38(11):14136–14147.

[25] Robert F. Bordley , Stephen M. Pollock. Assigning resources and targets to an organization's activities. European Journal of Operational Research. 1 August 2012; 220(3): 752–761.

[26] Pfaffenberger R.C, Walker D.A. Mathematical Programming for Economics and Business. first ed., The Iowa State University Press ;1976.

[27] Gass S.I. Linear Programming. fifth ed.,McGraw-Hill Book Company;1984.

REPORT OF INVENTIONS AND SUBCONTRACTS

(Pursuant to "Patent Rights" Contract Clause) (See Instructions on back)

Form Approved
OMB No. 9000-0095
Expires Jan 31, 2008

PLEASE DO NOT RETURN YOUR COMPLETED FORM TO THE ABOVE ORGANIZATION. RETURN COMPLETED FORM TO THE CONTRACTING OFFICER.

1.a. NAME OF CONTRACTOR/SUBCONTRACTOR	c. CONTRACT NUMBER	2.a. NAME OF GOVERNMENT PRIME CONTRACTOR	c. CONTRACT NUMBER	3. TYPE OF REPORT (X one)
Clark Atlanta University	W911NF-12-2-0067			a. INTERIM [ ] b. FINAL [X]
b. ADDRESS (Include ZIP Code)	d. AWARD DATE (YYYYMMDD)	b. ADDRESS (Include ZIP Code)	d. AWARD DATE (YYYYMMDD)	4. REPORTING PERIOD (YYYYMMDD)
223 James P. Brawley Drive, SW Atlanta, Georgia 30314	20120924			a. FROM 20120924 b. TO 20150923

SECTION I - SUBJECT INVENTIONS

5. "SUBJECT INVENTIONS" REQUIRED TO BE REPORTED BY CONTRACTOR/SUBCONTRACTOR *(If "None," so state)*

NAME(S) OF INVENTOR(S) (Last, First, Middle Initial) a.	TITLE OF INVENTION(S) b.	DISCLOSURE NUMBER, PATENT APPLICATION SERIAL NUMBER OR PATENT NUMBER c.	ELECTION TO FILE PATENT APPLICATIONS (X) d.				CONFIRMATORY INSTRUMENT OR ASSIGNMENT FORWARDED TO CONTRACTING OFFICER (X) e.	
			(1) UNITED STATES		(2) FOREIGN			
			(a) YES	(b) NO	(a) YES	(b) NO	(a) YES	(b) NO
None	None	None						

f. EMPLOYER OF INVENTOR(S) NOT EMPLOYED BY CONTRACTOR/SUBCONTRACTOR		g. ELECTED FOREIGN COUNTRIES IN WHICH A PATENT APPLICATION WILL BE FILED	
(1) (a) NAME OF INVENTOR (Last, First, Middle Initial) None	(2) (a) NAME OF INVENTOR (Last, First, Middle Initial)	(1) TITLE OF INVENTION	(2) FOREIGN COUNTRIES OF PATENT APPLICATION
(b) NAME OF EMPLOYER None	(b) NAME OF EMPLOYER		
(c) ADDRESS OF EMPLOYER (Include ZIP Code)	(c) ADDRESS OF EMPLOYER (Include ZIP Code)		

SECTION II - SUBCONTRACTS *(Containing a "Patent Rights" clause)*

6. SUBCONTRACTS AWARDED BY CONTRACTOR/SUBCONTRACTOR *(If "None," so state)*

NAME OF SUBCONTRACTOR(S) a.	ADDRESS (Include ZIP Code) b.	SUBCONTRACT NUMBER(S) c.	FAR "PATENT RIGHTS" d.		DESCRIPTION OF WORK TO BE PERFORMED UNDER SUBCONTRACT(S) e.	SUBCONTRACT DATES (YYYYMMDD) f.	
			(1) CLAUSE NUMBER	(2) DATE (YYYYMM)		(1) AWARD	(2) ESTIMATED COMPLETION
None	None				None		

SECTION III - CERTIFICATION

7. CERTIFICATION OF REPORT BY CONTRACTOR/SUBCONTRACTOR *(Not required if: (X as appropriate))* [ ] SMALL BUSINESS or [ ] NONPROFIT ORGANIZATION

I certify that the reporting party has procedures for prompt identification and timely disclosure of "Subject Inventions," that such procedures have been followed and that all "Subject Inventions" have been reported.

a. NAME OF AUTHORIZED CONTRACTOR/SUBCONTRACTOR OFFICIAL (Last, First, Middle Initial)	b. TITLE	c. SIGNATURE	d. DATE SIGNED
Johnson, Carol E.	Asst. V.P. for Research, Sponsored Progms. & Dual Degree Engineering	Carol E. Johnson	04/27/2016